A Mixed Reality Interface for Digital Twin Based Crane

Xinyi Tu

School of Engineering

Thesis submitted for examination for the degree of Master of Science in Technology.

Espoo 15.10.2020

Supervisor

Prof. Kari Tammi

Advisor

Juuso Autiosalo, Pauli Salminen

Aalto University
School of Engineering

Aalto University
School of Engineering

Aalto University, P.O. BOX 11000, 00076 AALTO
www.aalto.fi
Abstract of the master's thesis

Author Xinyi Tu

Title A Mixed Reality Interface for Digital Twin Based Crane

Degree programme ENG25 Master's Programme in Mechanical Engineering (MEC)

Major **Code of major**

Supervisor Prof. Kari Tammi

Advisor Juuso Autiosalo, Pauli Salminen

Date 15.10.2020 **Number of pages** 90 **Language** English

Abstract

With digitalization transforming manufacturing, we have been witnessing the "Industry 4.0" revolution, which is featured by "digital twin", the digital representations of the physical assets, processes or systems. The motivation of this thesis comes from tackling the challenges of designing a human-machine interface (HMI) for complex digital twin systems carrying with an increasing amount of data. As a case study, the work targets on prototyping and evaluating a proof-of-concept HMI leveraging state-of-the-art mixed reality (MR) technologies to enhance the operation of a digital twin based smart crane.

In this application context, the work firstly determines the hardware and software setup, then implements and evaluates the prototype - an MR application consisting of visualization, interaction, communication and registration modules. Those modules enable the application to function as a bridge of the bi-directional information exchange between the crane's digital twin platform (through the crane GraphQL server) and the user (through the HoloLens interface). In one direction, crane data is handled and displayed in a virtual dashboard in front of the users' view for monitoring; In the other direction, users can navigate the crane by interacting with holograms via either fixed or movable target control method, with great flexibility and mobility ensured by spatial tracking and registration. The prototype is then quantitatively evaluated regarding the accuracy of both control methods, with the testing data well visualized through error ellipsoids, which indicates the fixed target control method excels the movable one by lower variance. With wrapping up the work, six concrete research topics for further development of the MR application are proposed and elaborated.

Keywords Mixed Reality, Digital Twin, HoloLens, Smart Cranes, Human Machine Interface, Industrial Communication, Spatial Tracking and Registration

Contents

Abstract 3

Contents 4

Abbreviations 6

1 Introduction 7
- 1.1 Motivation . . . 7
- 1.2 Research Environment . . . 7
  - 1.2.1 Aalto Industrial Internet Campus (AIIC) . . . 7
  - 1.2.2 MACHINAIDE Project . . . 8
- 1.3 Research Problem and Questions . . . 9
- 1.4 Objectives and Scope . . . 9
- 1.5 Thesis Structure . . . 10

2 Background 12
- 2.1 Industry 4.0 and Digital Twin . . . 12
  - 2.1.1 Industry 4.0 . . . 12
  - 2.1.2 Digital Twin . . . 14
- 2.2 Human Machine Interaction and Mixed Reality . . . 21
  - 2.2.1 Human Machine Interaction . . . 21
  - 2.2.2 Mixed Reality . . . 23
- 2.3 Industrial Communication and Distributed Systems . . . 27
  - 2.3.1 OPC UA . . . 28
  - 2.3.2 REST and GraphQL . . . 29
  - 2.3.3 Parallel Computing and Multi-threading . . . 32
- 2.4 Tracking and Registration . . . 33
  - 2.4.1 Tracking Systems . . . 34
  - 2.4.2 Spatial Relationship Graphs (SRG) and Patterns (SRP) . . . 35
  - 2.4.3 Calibration and Registration . . . 39

3 Hardware Setup 41
- 3.1 Microsoft HoloLens (1st Generation) . . . 41
  - 3.1.1 HoloLens Overview . . . 41
  - 3.1.2 Device Specifications . . . 41
  - 3.1.3 Device Capabilities . . . 43
- 3.2 Smart Crane “Ilmatar” and AIIC . . . 46
  - 3.2.1 Crane Overview . . . 47
  - 3.2.2 Crane Movement Subsystems . . . 48
  - 3.2.3 Crane’s Operation Environment: AIIC . . . 48

4 Software Setup 51
- 4.1 Software System and Connectivity of the Crane . . . 51
  - 4.1.1 Crane Software System Overview . . . 51

4.1.2 Crane OPC UA Interface and Data Access 51
4.1.3 Crane OPC UA - GraphQL Wrapper 53
4.1.4 Crane Connectivity 56
4.2 Software and SDKs for MR Development 57
4.2.1 Unity 3D Engine and Mixed Reality Toolkit (MRTK) 57
4.2.2 Vuforia Augmented Reality SDK 59
4.2.3 RestSharp Library 62
4.2.4 Task Parallel Library (TPL) in .NET Framework 62

5 Prototypical Development **63**
5.1 Prototype Architecture Overview 63
5.2 MR Application Modular Design 64
5.2.1 Interaction Module 64
5.2.2 Visualization Module 66
5.2.3 Registration Module 69
5.2.4 Communication Module 73

6 Evaluation **76**
6.1 Evaluation Procedure 76
6.1.1 Collecting Evaluation Data 76
6.1.2 Calculating Control Accuracy 76
6.1.3 Visualizing Evaluation Data 76
6.2 Evaluation Result 77
6.2.1 Collected Data and RMSE Calculation 77
6.2.2 Error Ellipsoid Visualization 78

7 Conclusion and Outlook **79**
7.1 Answers to Research Questions 79
7.1.1 Q1: Prototyping - Challenges and Solutions 79
7.1.2 Q2: Evaluation - Challenges and Solutions 82
7.2 Limitations and Future Work 83

References **86**

Abbreviations

AI	Artificial Intelligence
API	Application Program Interface
AR	Augmented Reality
CPU	Central Processing Unit
CS	Coordinate Systems
DFG	Data Flow Graph
DFN	Data Flow Network
DOF	Degrees of Freedom
DT	Digital Twin
FOR	Frames of Reference
GraphQL	Graph Query Language
HMD	Head-Mounted Display
HMI	Human Machine Interaction / Interface
IIoT	Industrial Internet of Things
IMU	Inertial Measurement Unit
IoT	Internet of Things
MR	Mixed Reality
MRTK	Mixed Realty Toolkit
MVEE	Minimum Volume Covering Ellipsoid
M2M	Machine to Machine
NLP	Natural Language Processing
Ilmatar OIE	Ilmatar Open Innovation Environment
OPC UA	Open Platform Communications Unified Architecture
PLC	Programmable Logic Controller
PLM	Product Lifecycle Management
POC	Proof-of-Concept
RMSE	Root Mean Square Error
SDK	Software Development Kit
SRG	Spatial Relationship Graph
SRP	Spatial Relationship Pattern
TPL	Task Parallel Library
WMR	Windows Mixed Reality
VR	Virtual Reality
UI	User Interface
UWP	Universal Windows Platform
UX	User Experience

1 Introduction

1.1 Motivation

Nowadays, digitalization is fundamentally and sustainably transforming the industry world. The industrial production process is moving towards a new dimension in the "Industry 4.0" and Internet of Things (IoT) innovations, which is characterized by introducing digital twin, the digital representations of the physical assets, processes or systems.

On one hand, digital twin could assist in machine design, system simulation, product life cycle management (PLM), performance optimization, predictive maintenance and new service model development with high flexibility and efficiency.

On the other hand, the increasing amount of digital twin data within different formats and from different resources creates a challenging context for human-machine interaction (HMI) when safety and situational awareness are essential. Highly supportive and intuitive HMI used for operating machines and managing services are severely lacking. Comprehensive research is therefore needed, on how to refine the information, highlight relevant content, manage multiple access points and explore innovative and inclusive interaction methods that simplify operations, enhance safety, correct decision and consequently lead to high acceptance and value for users.

Therefore, the motivation for this thesis can be found from the challenges of designing a human-machine-interface to handle and visualize the increasing amount of data brought by the increasing digital twin applications in nowadays's industry world.

1.2 Research Environment

Several European initiatives are playing central roles heading towards the direction of digital twin application development. Under this trend, the thesis work is conducted at Aalto Industrial Internet Campus (AIIC) and as part of the MACHINAIDE project work package.

1.2.1 Aalto Industrial Internet Campus (AIIC)

The researchers at the Aalto Industrial Internet Campus (AIIC) are studying the digital twin concept and its potential, with the aim of developing methods and solutions that can benefit the Finnish industry as a whole in their digital transformation. The development concretizes in joint piloting and hands-on work, especially around the Ilmatar Open Innovation Environment (OIE), with a focus on the coupling of the crane and its digital twin with product configuration, product design, and product life cycle.

Ilmatar OIE is an open physical and digital development environment targeted for different third parties i.e. students, startups, SMEs, larger corporations or other parties, who want to innovate and develop new devices and applications that are connected to Konecranes overhead smart cranes equipped with extensive digital systems. It includes a physical crane environment located at AIIC premises and multiple software components to support development. The environment enables fast cyber-physical prototyping with solutions from the latest research activities, being even further than current state-of-the-art industry solutions.

The thesis work will conduct experiments on Ilmatar OIE at AIIC as a case study, utilizing its existing available applications and resources related to digital twin based services, and will eventually contribute to the constantly evolving environment by its resulting demos and applications.

1.2.2 MACHINAIDE Project

The thesis is under the framework of the MACHINAIDE project, an international EUREKA ITEA3 project with 17 participants from Finland, The Netherlands, Turkey and Korea. The international project is coordinated by Konecranes Global. Participants from Finland also include Aalto University, Ideal PLM, Remion, RollResearch International and VTT Technical Research Center of Finland.

Machine builders are collecting data related to their products within different formats and tools for several years. The number of related digital twins increases each time, products of other manufacturers become included in the own one, or in case of acquiring other companies. For coping with this development, the MACHINAIDE project is exploring the possibilities of creating new innovative activities and services created by the interaction of digital twin devices and the accumulation of information.

The research will determine how data produced by digital twins in multiple ecosystems can be effectively combined. In addition, options for future business models are being explored. The business is expected to come from managing digital twin data and providing value-added information to multiple parties. The aim is to increase our understanding of the ecosystem platform and value creation required by digital twins, to support innovative concepts for accessing, searching, analysing and using multiple digital twins data for the major purpose of increasing usability and functional upgrading of machines and equipment within the crane and printing machine domains.

Overall, the thesis work shares the motivation and objective of the MACHINAIDE project. Together with other researchers from Aalto University, we also aim at strengthening Aalto University's Industrial Internet capabilities and research on the potential of the digital transformation by leveraging the strong international dimension brought by the project. The research will be continued by expanding the use of the digital twin to new applications, communication interfaces and user interfaces. The applications build innovative user interfaces between machines and humans with the target of making product use and life cycle management more efficient and smoother.

1.3 Research Problem and Questions

Based on the motivation and research environment described above, the research problem and questions are determined as follows.

Research Problem

How can an innovative HMI utilizing MR Technologies enhance digital twin based smart crane operation?

Research Questions

The research problem can be further divided into the following two concrete research questions:

1. How to develop a proof-of-concept (POC) prototype of an MR interface for digital twin based crane operation? <discussed in chapter 5>
2. How to evaluate the performance of the MR interface in terms of the interaction accuracy with different control methods? <discussed in chapter 6>

1.4 Objectives and Scope

Overall, this thesis work aims at developing an innovative HMI leveraging MR technologies to enhance the operation of a digital twin based smart crane “Ilmatar” as a case study, which is divided into two main objectives, MR interface prototyping and evaluation.

In order to guarantee the software’s re-usability, the project’s architecture is designed in a modular manner, meaning that different components hold a clear focus on different functionalities like visualization, controlling, calibration and communication.

Along with the prototyping process, the work also aims at exploring and exploiting the capability of the software and hardware in use of the work, since the software and hardware setups are limited to the resources that are currently available within the research environment or the development tools that are common practice or the most recently released. It is worth pointing out that, MR is such an emerging technology that the fast updating of hardware, the development tools and packages also bring with the compatibility issue across them both in horizontal and vertical directions, which need to be resolved in order to integrate different components in one coherent interface. Regarding evaluation, the thesis does not cover the user experience survey on the final prototype due to the COVID-19 epidemic situation. Instead, it quantitatively evaluates the interaction performance through accuracy measurements with different control methods available in the work.

1.5 Thesis Structure

To begin with, the thesis introduces the motivation and the research environment. Based on those, it then raises the research problem and questions, followed by the objectives and scope correspondingly. Then it comes to this section about the thesis structure.

In the background chapter, the thesis first explains the concept of the digital twin in industry 4.0 context to a sufficient level for understanding the work's relevance to it. Secondly, it screens general state-of-the-art HMI technologies and applications with highlights on the concept, principles and features of related MR technologies, as those can be candidates for introducing HMI innovations to digital twin based services. Thirdly, an overview of industrial communication and networking related technologies are provided, followed by an introduction on the distributed system technologies like parallel computing and multi-threading, as those are critical for the communication between MR application with crane digital twin platform. Fourthly, the tracking, calibration and registration techniques in AR/MR applications are explained, as the adoption of those can introduce great flexibility to the user experience of the application.

In the chapters of hardware and software setup, the MR device (i.e. Microsoft HoloLens 1st generation), platform resources (i.e. the smart crane "Ilmatar" and AIIC with its software systems), as well as development tools for MR applications (i.e. Unity3D Engine, MRTK, Vuforia SDK, RestSharp, and Task Parallel Library in .NET Framework) in use of the work are described with core features and functionalities highlighted.

In the prototyping chapter, it starts with depicting the architecture of the POC prototype at a high level, then highlights the modular design of the MR application, with a detailed explanation of each module's functionalities from users' perspectives, as well as the principle and logic behind the scene.

The following evaluation chapter illustrates a detailed procedure of measuring, calculating and visualizing the prototype performance in terms of the control accuracy with different control methods. The collected data, computed errors and generated plots are presented to demonstrate the quantitative evaluation result.

With wrapping up and condensing the results, summative answers to both research questions are given, which lead to a high-level discussion. An outlook of the thesis work comes at the end of the thesis.

Figure 1 shows the structure of the work.

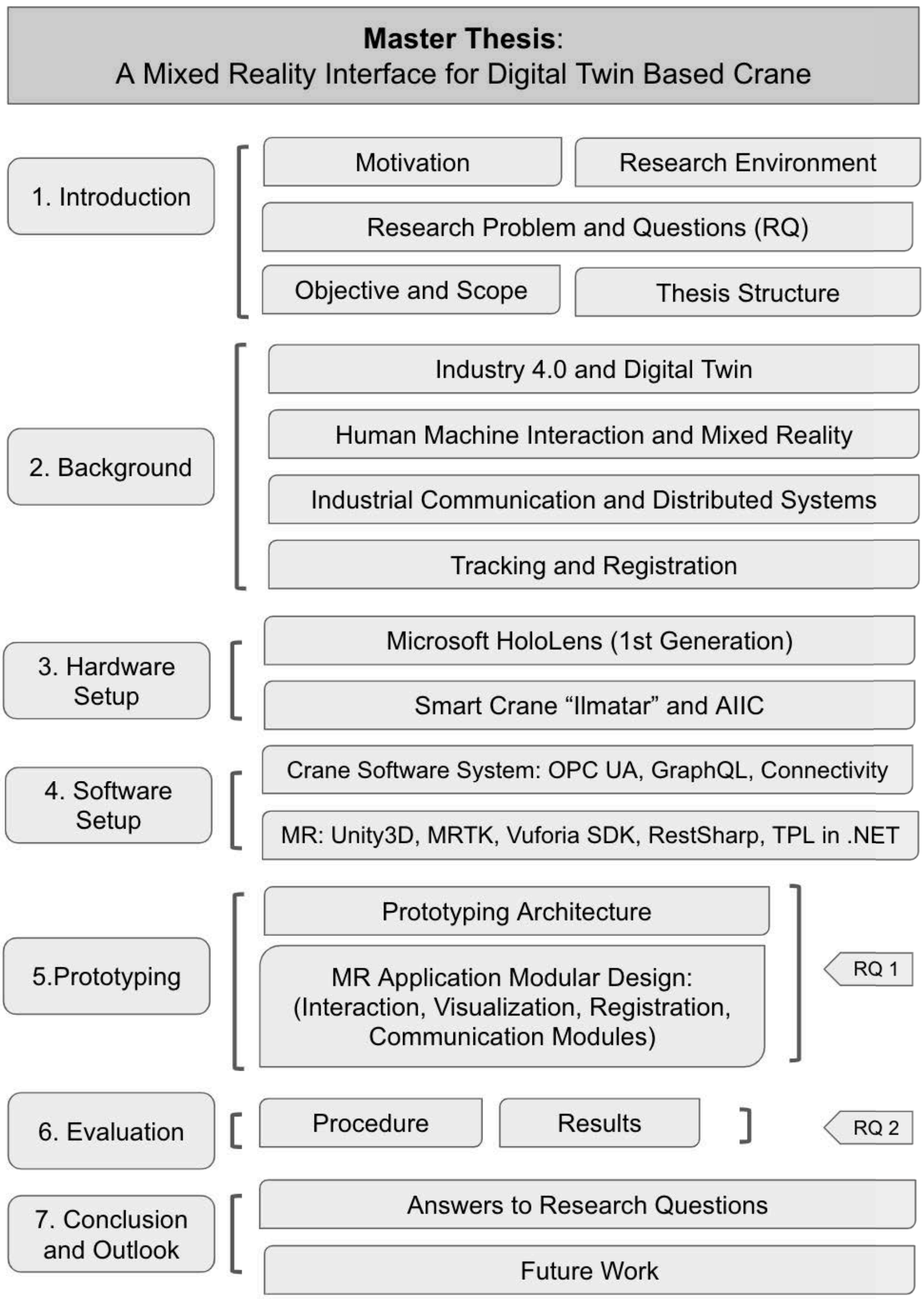


Figure 1: Structure of the Work

2 Background

In this chapter, the background information with a literature review of four different fields will be presented, as they are directly related to either the use case context or the functionality components forming the thesis work, i.e. digital twin, mixed reality, industrial communication, and registration/tracking techniques in AR/MR.

2.1 Industry 4.0 and Digital Twin

This section explains the history and concept of Industry 4.0, industrial internet of things (IIoT) and digital twin, as well as digital twin in Industry 4.0 context to a sufficient level for understanding the work's relevance to it.

2.1.1 Industry 4.0

Background of Industry 4.0

From manual labor to mechanization to electrification to digitalization, throughout the years from the 1800s until today, technological innovations and inventions have been giving rise to every industrial revolution, with the complexity and productivity of the processes or systems increasing simultaneously and dramatically.

Since the third industrial revolution, also known as the digital revolution, which is focused on electronic systems, IT systems and automation, extraordinary development has been supervened in numerous fields from information, computer to communication technologies. With these modern technologies being adopted in the current manufacturing industry landscape, which enabled accessible integration of interconnected intelligent components in the industrial process, the fourth transformation has been triggered [32].

This trend was first termed as "Industrie 4.0" in 2011 by a group of representatives from various fields of business, politics, and academia with the initiative to boost the German competitiveness in manufacturing. In 2014, the term became widely acknowledged due to the call for related research with a huge amount of funding by the German government ministry, which was followed by further adoption of the idea in the German High-Tech Strategy for 2020. Subsequently, a Working Group was formed to further advise on the implementation of Industry 4.0. At the same time, similar concepts were being put forward on a global scale, such as 'Industrial Internet' by General Electric in the USA [7].

A report from the Boston Consulting Group [46] describes the nine fundamental technology trends that function as the building blocks of Industry 4.0 (see Figure 2).

Even though most of these technologies forming the Industry 4.0 foundation have already been adopted independently in manufacturing, Industry 4.0 enables them not only to transform the production flow with greater cost-effectiveness and efficiency, but also to shift the production relationships among suppliers, producers, and customers,

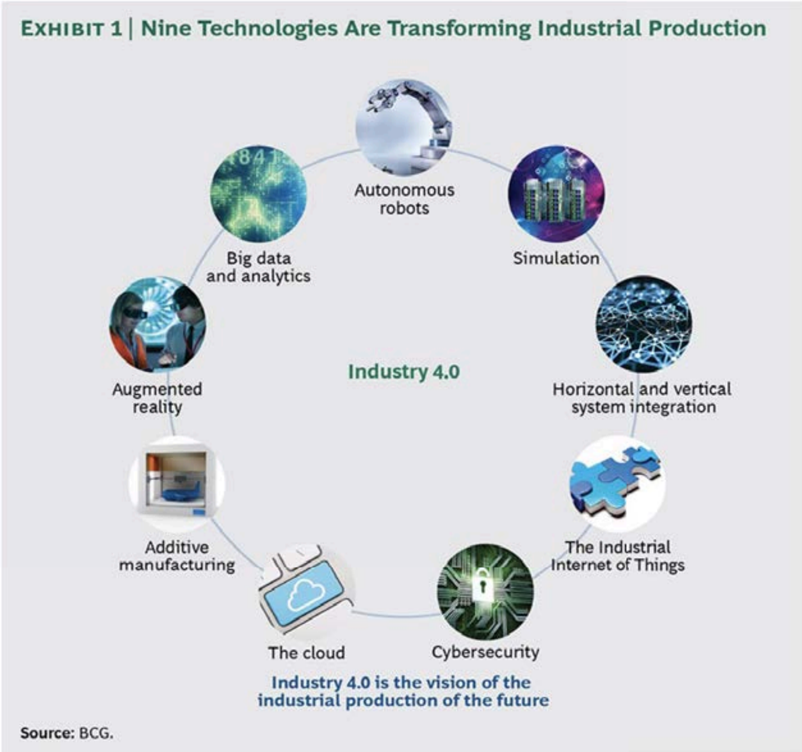


Figure 2: Nine Technologies Tranforming Industrial Production [46]

along with those between human and machines, by providing a fully integrated, automated and optimized platform-based approach.

The integration of these technologies has been empowered by efficiency and cost-effectiveness into even smaller products and components, which enables them to be aware of the status of their own, of the environment and the process, as well as to exchange and share the data with external systems [14]. Product and production data from different sources, such as equipment sensors, enterprise and customer management systems, is accessible by external systems as well as authorized users, collected and stored in the database, and utilized for further analysis, evaluation, monitoring, .etc [5].

Industrial Internet of Things (IIOT)

The internet of things (IoT) is basically a system of interrelated electronics, sensors, software, and networking capabilities embedded in physical devices and products ("things"), such as computing devices, mechanical and digital machines [32]. An IoT system usually brings with it unique identifiers that allow communicating data through a network and are free of human-to-computer or -human interaction.

The industrial internet of things (IIoT) refers to the subset and extension of the IoT in industrial sectors and applications. In the context of the industrial world, various components embedded in machines typically enable the communication among them and with their central controller systems through networks under standard protocols like Rest, OPC UA, SOAP, MQTT, .etc. In such a system, data collection, analytics, evaluation and data-based decision making are therefore decentralized [46].

As a result, the IIoT empowers enterprises and industries with greater reliability, accuracy and efficiency in their operations and thus bigger economic benefits [32], utilizing machine-to-machine communication, artificial intelligence, big data, networking and cloud technologies. Various industrial applications are covered by the IIoT, such as medical devices, robotics products, and software-specific production systems and processes. Such kinds of products could be termed as connected smart products as well, according to Harvard Business Review [38].

Related research has been conducted, for instance, on data acquisition of both sensory and manufacturing resources from IIoT-empowered CNC machines [5] (See Figure 3).

Figure 3: IIoT-enabled CNC Machine Data Acquisition [5]

2.1.2 Digital Twin

Background of Digital Twin

The ”digital twin” was brought up from the early ideas in early 2002, where “a conceptual idea” was proposed for Product Lifecycle Management (PLM). The concept contains three fundamental components: the virtual and physical space, as well as the constant information exchange between them (See Figure 4). The three components build up the digital twin, which is characteristic of real-time data communication between the physical world and its digital counterpart [12].

As the early conception of the digital twin was mainly meant to serve the PLM, the representation of the real world by its virtual counterpart, therefore, plays a role all through a product’s lifespan, i.e. development and introduction, growth, maturity/stability, and decline.

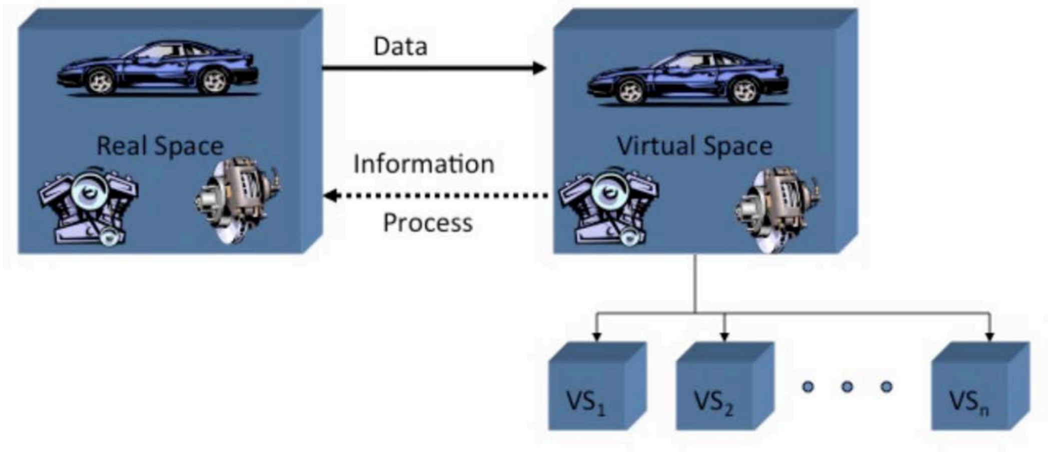


Figure 4: Conceptual Ideal for PLM [12]

A similar concept proposed by [9] in 2003 is a system composed of a physical product and its digital counterpart model. Along with the lifecycle of the product, the digital counterpart can exchange data requested by external systems with the physical product in a real-time manner, thanks to the advances in IIoT and network technologies in the manufacturing industry by then [9].

The data exchange model between the physical system and its digital counterpart is then titled an "information mirroring model" in 2006 [12].

At the beginning of Industry 4.0 in 2012, the term "digital twin" was first adopted in 2012 by NASA, where the digital twin is envisioned in its space vehicles as part of simulation-based system engineering. In detail, the digital twin is conceptualized to facilitate the probabilistic simulation of NASA's aerospace vehicles, and the integration of real-time data from different sensors and historical data mainly for maintenance. This digital twin is also argued to benefit in reducing potential damage as it can propose a modification in the mission profile so that the probability, as well as the lifespan of the mission success, can be improved [47].

Definition of Digital Twin

The definition shift of digital twin in the publications in academia and business domain along the years from 2012 to 2016 is derived from the Scopus database and summarized by [32]. (See Figure 5)

As mentioned in the previous section, NASA first defines the digital twin in the context of its space vehicles simulation in 2012. The following years then witness the expansion of this term into an increasingly wider variety of domains and contexts with different applications and objectives. How to define and perceive the digital twin is highly reliant on its functionalities and capabilities. In some cases, the digital twin is solely used for the simulation and modeling of physical assets, systems or processes. In other cases, it is also adopted for predictive maintenance or future design improvement by monitoring and evaluating e.g. the inefficiency or stress load of machinery [37]. Despite the variety of objectives and applications, the basic concept composing a digital twin stays persistent.

No.	Year	Definition
1	2012	An integrated multi-physics, multi-scale, probabilistic simulation of a vehicle or system that uses the best available physical models, sensor updates, fleet history, etc., to mirror the life of its flying twin. The digital twin is ultra-realistic and may consider one or more important and interdependent vehicle systems.
2	2012	A cradle-to-grave model of an aircraft structure's ability to meet mission requirements, including sub models of the electronics, the flight controls, the propulsion system, and other subsystems.
3	2012	Ultra-realistic, cradle-to-grave computer model of an aircraft structure that is used to assess the aircraft's ability to meet mission requirements.
4	2013	Coupled model of the real machine that operates in the cloud platform and simulates the health condition with an integrated knowledge from both data driven analytical algorithms as well as other available physical knowledge.
5	2013	Ultra-high-fidelity physical models of the materials and structures that control the life of a vehicle.
6	2013	Structural model which will include quantitative data of material level characteristics with high sensitivity.
7	2015	Very realistic models of the process current state and its behavior in interaction with the environment in the real world.
8	2015	Product digital counterpart of a physical product.
9	2015	Ultra-realistic multi-physical computational models associated with each unique aircraft and combined with known flight histories.
10	2015	High- fidelity structural model that incorporates fatigue damage and presents a fairly complete digital counterpart of the actual structural system of interest.
11	2016	Virtual substitutes of real-world objects consisting of virtual representations and communication capabilities making up smart objects acting as intelligent nodes inside the internet of things and services.
12	2016	Digital representation of a real-world object with focus on the object itself.
13	2016	The simulation of the physical object itself to predict future states of the system.
14	2016	Virtual representation of a real product in the context of Cyber-Physical Systems.
15	2016	An integrated multi-physics, multi-scale, probabilistic simulation of an as-built system, enabled by Digital Thread, that uses the best available models, sensor information, and input data to mirror and predict activities/performance over the life of its corresponding physical twin.
16	2016	A unified system model that can coordinate architecture, mechanical, electrical, software, verification, and other discipline-specific models across the system lifecycle, federating models in multiple vendor tools and configuration-controlled repositories

Figure 5: Definition of Digital Twin in Scopus Study [32]

Types of Digital Twin

As proposed by [12], digital twins can be categorized based on their manifestation nature into digital twin prototype (DTP), digital twin instance (DTI), digital twin aggregate (DTA) and digital twin environment (DTE).

DTP presents a physical product at its prototypical phase accompanying datasets with different facets, such as design, components, processes and services, .etc., which serve to produce the product.

DTI presents a physical product constantly connected with data generated alongside its lifecycle. Empowered by sensor technology and others, the datasets possibly access all kinds of facets, not only those in DTP but also maintenance and service-related operational status, in the form of either historical, real-time or predicted future information.

DTA presents an aggregation of DTIs, the links and data flow among them. Data exchange can happen within DTA based on proactive queries or ad-hoc forms. Information across all the DTI units can be aggregated within DTA to generate a united profile.

DTE presents a multi-domain, combined physical application space, which operates the digital twin products of different types as mentioned above. Among all the DTEs with different usages, interrogative DTEs are adopted to fetch historical or real-time operational status, while predictive DTEs are employed to predict future system or process behaviors for maintenance purposes or validate whether a new design or recent update would satisfy the prerequisites.

Digital Twin in Industry 4.0

As discussed in the previous section, industry 4.0 is characteristic with digitalization and inter-connectivity in the smart factory environment, where data is exchanged, aggregated, analyzed and processed to enable highly automated and intelligent behaviors in the physical processes and systems [37]. In such a context, a digital twin can function as a platform that hosts various emergent technologies featuring Industry 4.0. This way, the digital twin can boost the design conception, production process, system operations and predictive maintenance.

A feature-based digital twin framework formulated in literature studies is proposed [17] (see figure 6). The framework includes nine emergent technologies, in other words, technical features, and the integration of them through connection with the data link in the center. The features, i.e. computation, coupling, identifier, security, data storage user interface, simulation model, analysis, and artificial intelligence, mark the functional requirements in implementing a digital twin in industry 4.0.

Similarly, a digital twin model (See Figure 7) and a conceptual reference architecture for applying digital twin in Industry 4.0 (see Figure 8) are depicted [37]. The model is an integration of the physical and digital components with data flow and PLM loop from acting (creating, communicating, aggregating and analyzing) to insight.

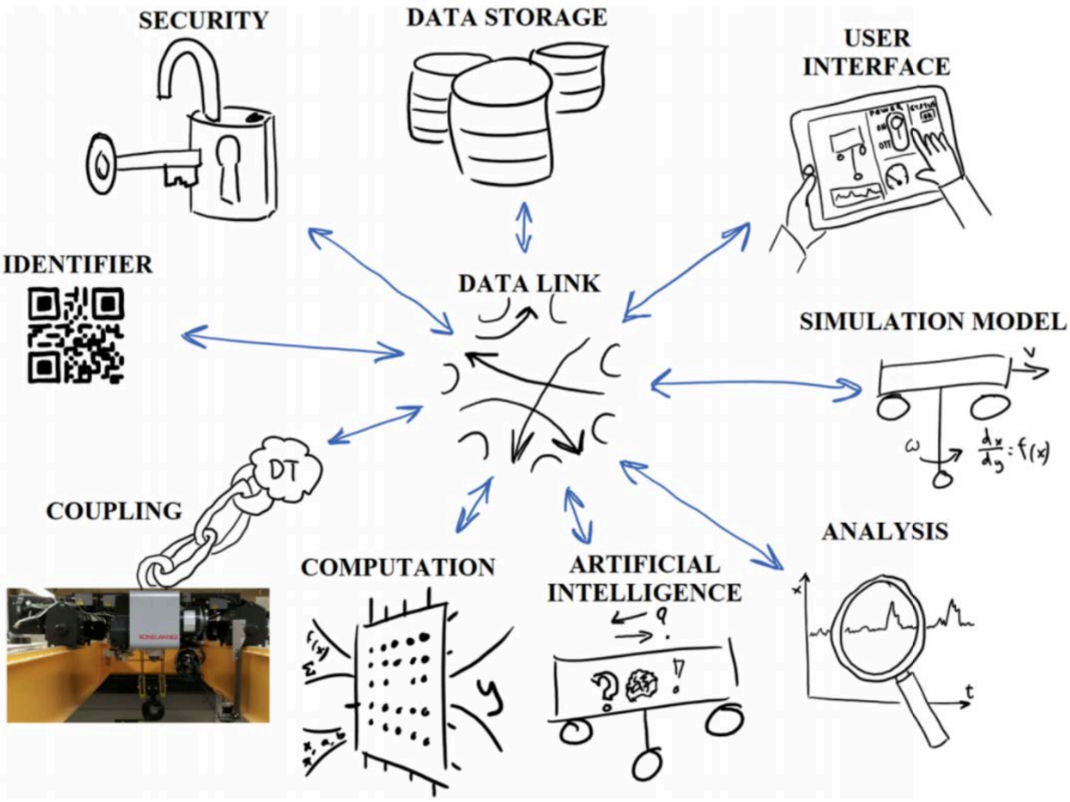


Figure 6: Feature-based Digital Twin Framework [17]

Among all the building components typically forming a digital twin system in Industry 4.0 context, there are three main technical features of importance in implementing digital twins, i.e sensors and actuators, data-related technologies, integration, and user interfaces.

Sensors and actuators play a particularly critical role in the digital twin model, as they are the fundamental elements of any IoT system. Sensors embedded in the machinery empower the system to access and collect data from working assets, operational processes or the environment, such as pressure, temperature, flow, .etc. Actuators, on the other hand, empower users or the computer algorithms to act (control or intervene)on the processes or the system behaviors.

Data-related technologies empower the information flow of a digital twin model in Industry 4.0, as they ensure the historical, real-time and predictive future status of the physical components or processes preserved throughout the product lifecycle. Data within digital twin systems can incorporate various types of information, like previous or current sensor data, enterprise resource planning (ERP) system data, manufacturing execution systems (MES) data, CAD models or simulation data, legacy data, .etc. It is typically collected, aggregated and preserved through different data ingestion procedures into data lakes. then handled utilizing various data management and big data technologies. Different tricks for data analytics, such as

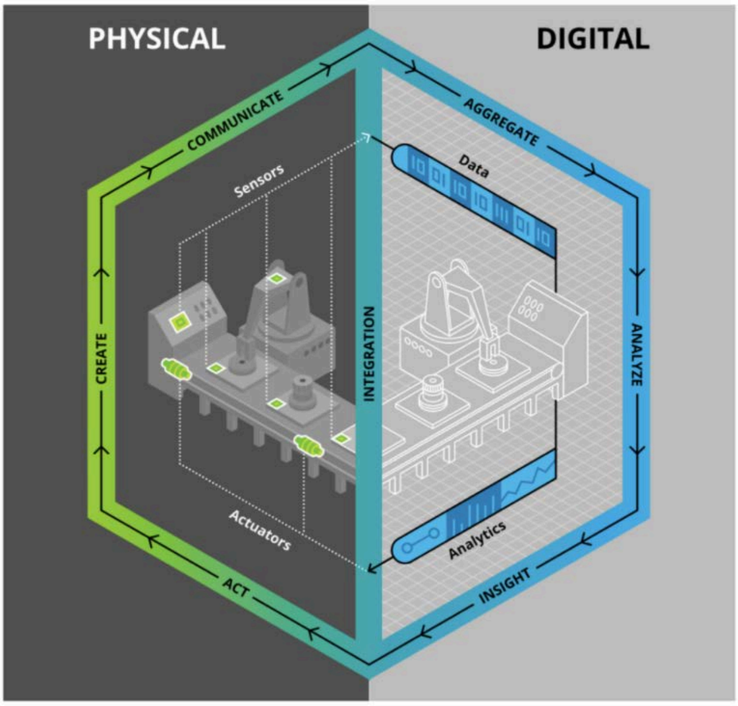


Figure 7: Digital Twin Model [37]

artificial intelligence, cognitive engines, hybrid models, can be utilized to provide insights out of the data, which could benefit the predictive maintenance or boost the system update and process optimization and future design.

Integration illustrates the networking technologies and standard protocols that serve various functionalities like monitoring, coupling, communication. It usually provides interfaces between physical components and their digital counterparts, or between the system itself and external devices. Identification of both the physical and digital content in an interconnect environment is critical for integration out of security and privacy reasons, as data should be neither accessed nor manipulated by unauthorized devices. Common identification methods are like RFID tags and IP addresses.

User interfaces define how operators interact with the digital twin systems through devices ranging from tablets, smartphones to overhead displays and smart glasses. With user interfaces, operators are able to, on one hand, monitor the process status or system behaviors by fetching and perceiving the data visualized in the display devices, and getting notifications for special events. On the other hand, operators can control and manipulate the physical components through user interfaces by touch, voice, gesture-based interaction, typically empowered by advanced technologies such as mobile, web and AR/MR/VR in a modern smart factory environment.

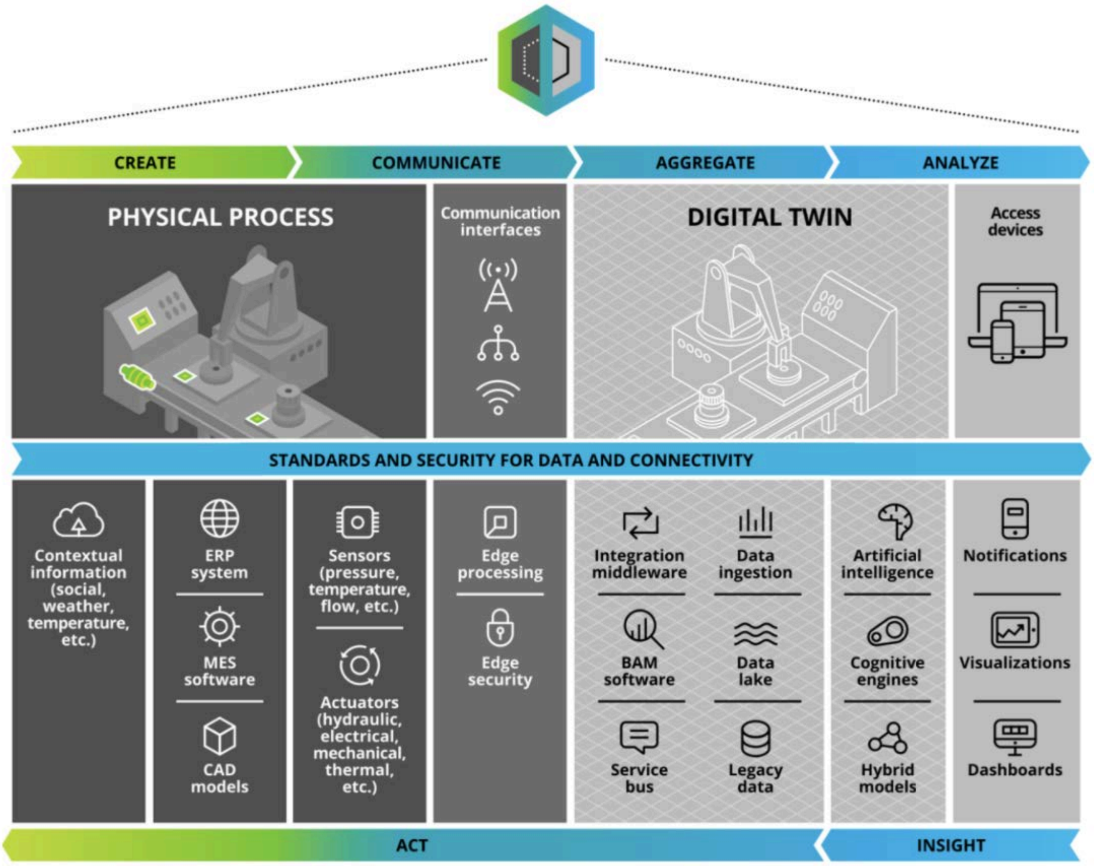


Figure 8: Conceptual Reference Architecture for Digital Twin in Industry 4.0 [37]

2.2 Human Machine Interaction and Mixed Reality

This section provides an overview of HMI technologies and their applications in Industry 4.0 context, with screening the state-of-the-art and innovative HMIs. We then highlight the background, concepts and features of related MR technologies, as those can be candidates for introducing HMI innovations to digital twin based services.

2.2.1 Human Machine Interaction

HMI Overview

Human-machine interaction (HMI) is a field that studies the interface and interaction/communication ways between machines and its users, typically also taking the environment, where the users perform operations on the machine, into design consideration.

As a technology playing a critical role in smart manufacturing and process optimization, HMI consists of various components, i.e. interaction, communication, as well as cooperation among humans, machines and the operation environment.

Rich information exchange is expected to happen in a functional HMI to empower the machine users, typically utilizing any of the four types of interaction modes, i.e. image, voice, data and intelligence interaction [59].

One key criterion to consider while designing an HMI is the harmony of the interaction, which, in other words, means the reduction of the user's cognitive load, and the increase of the operational capability and productivity.

HMI can be adopted throughout the whole product lifecycle including product design, manufacturing, and service, where the primary concerns increasing the requirements on HMIs are efficiency and safety.

HMI in Industry 4.0

As illustrated in the previous section, industry 4.0 transformation is characterized by the joined communication and processing capabilities of each unit in one system, which leads to new requirements for designing HMI. In other words, with the increasing autonomy and complexity of a typical production process or manufacturing system within the Industry 4.0 context, the machine operators are also faced with challenges and possess expectations that are different from those before such a transformation.

One key design consideration of HMI in industry 4.0 is to integrate users into the architecture of the cyber-physical systems. To empower this, various types of mobile devices have started to be adopted in manufacturing and production, including overhead display, tablets, smartphones, etc., with multi-modal (gesture, voice, position) or multi-touch interaction capabilities. On the other hand, HMI should also enable or at least function as a gateway for visualization, aggregation and analysis of the rich data coming from different units within an IIoT system.

With a modern adapted production strategy applied in Industry 4.0, while the systems and processes are organized autonomously, the machine users are expected to perceive the information that is aggregated, prepared and well presented by HMI to monitor and if necessary give intervention to the processes [10].

Innovative HMI Types

The advances in information and communications technology (ICT) has contributed immensely to numerous fields, among other things, bringing innovation and revolution to HMI studies. New emergent technologies like web, mobile, big data, AI, digitalization, AR/MR, speech generation and recognition gradually take place in HMI activities, with wearable devices like wristbands, watches, overhead displays involved.

The evolution of HMI can be divided into four stages. The first stage solely relies on the screen display and keyboard input, while the second is based on graphic display and mouse input, although either of their interaction content is composed of texts and characters. At the third stage, touch screen, camera stream, and voice input started to be involved. Thanks to the advances in VR/AR/MR, gesture recognition, eye/head tracking, .etc., the fourth stage has witnessed the transformation of HMI towards a more intuitive and accessible manner. These new innovative HMI types applied in Industry 4.0 can be mainly categorized as touch, voice, gesture and AR/MR/VR interfaces.

Touch Interfaces: Since introduced in the 1990s, touch interfaces have been commonly adopted in HMI as they ease the interaction between users and machines with user-friendly data visualization and straightforward input mechanisms. Touch interface applications ease the machine-to-machine communication in IIoT and empower manufacturers to monitor and control operational processes from local or remote factories and locations with machinery equipment. Modern touch interfaces are featured with more safety and comfort for the operators, as they are usually more responsive and even support e.g. touching with gloves.

Voice Interfaces: Voice interfaces enable humans to "talk" with machines, without complex physical interaction but in a natural way as if the machines could listen, interpret, and speak. Voice interfaces are not necessarily equipped with screens to illustrate information, but they facilitate data access through intuitive, efficient and hands-free interactions. In the context of Industry 4.0, voice interfaces start playing a crucial role, especially when the remote operation is required.

Gesture Interfaces: Gesture interfaces are on trend as an alternative HMI modality as it empowers users to interact with machines in a more intuitive way. Gesture Interface enables touchless control of machines or computer systems by recognizing users' head, its position and movement, calculating and interpreting necessary info as inputs for the machinery systems accordingly.

AR/MR/VR Interfaces: AR/MR interfaces visualize the digital content into virtual objects and combine them with the real-world scene, typically in a real-time manner. AR/MR applications dramatically transform the operation environment and provide an immersive and interactive experience for users. VR is solely built on scenes/graphics generated by computers, with possibly enhanced details of the original models. The AR/MR/VR interfaces are thoroughly discussed in the following section.

2.2.2 Mixed Reality

Background of MR/AR/VR

The origin of augmented reality (AR) and virtual reality (VR) came way back in 1838. AR is defined as the extended reality where supplementary contextual computerized information is added to the reality and enriches the user's perception of the real environment [49]. VR is a related but distinct concept, where the user is fully immersed in a computerized virtual environment without any reference to the real world, meaning that even such as the fundamental physics law of gravity, time, and materials can be overridden and replaced those in the real world with the computerized ones [31].

Mixed reality (MR) is characterized by the combination of both AR and VR environments. It is acknowledged as the forthcoming evolution in the interactions among humans, computers and the environment (see Figure 9).

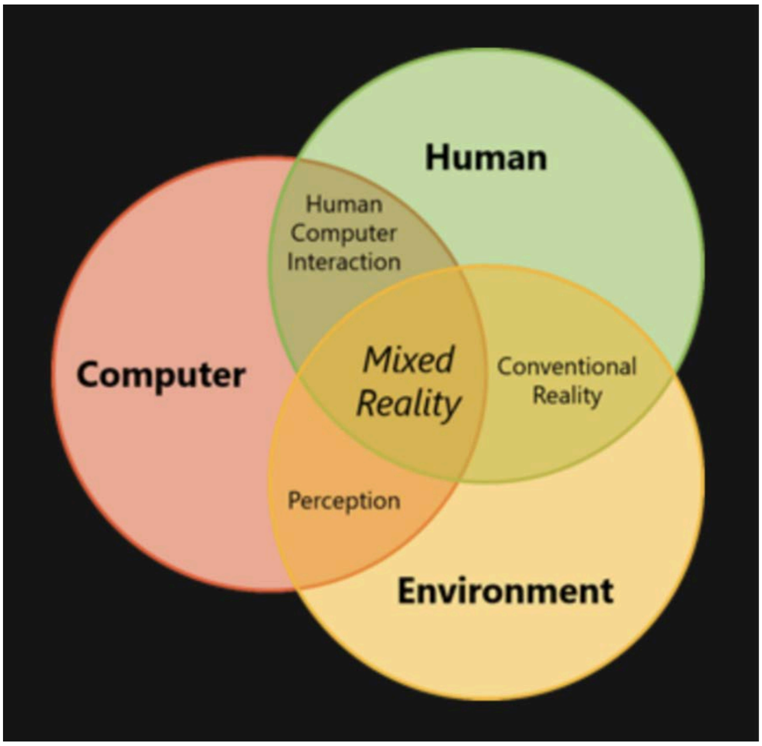


Figure 9: Human-Computer-Environment Interactions [48]

MR technologies offer new opportunities for developing a hybrid reality with digital content embedded in the real world. Empowered by the technological advances and developments in the fields of display, sensors, graphical processing power, input systems and computer vision, the possibilities for MR are boosting beyond our imagination [48].

The concepts of AR/ MR can be adopted interchangeably. AR is most frequently used in gaming, such as in the trendy mobile game "Pokemon Go". Nonetheless, the huge potential of this technology is foreseen in the industrial and manufacturing domain with numerous related studies going on. AR has been acknowledged as one of the top ten strategic technology trends of the year 2018 (Cearley David; Burke Brian, 2018) [6].

Reality-Virtuality (RV) Continuum

In the first place, AR was defined in a rather general manner and as a form of VR with transparent head-mounted display (HMD). However, it is questioned since the similarity and difference between the concepts of AR and VR proved to be more complex than what is stated in this definition.

VR aims to provide users an immersive experience inside a fill-in reality, which is totally under the control of its developer. The users typically only interact with virtual components, and there exists no object-tracking on the real world. The challenges to tackle in VR applications are like user's effective immersion and VR motion sickness. On the contrary, AR deals with enriching instead of replacing the existing reality as such in VR.

To distinguish VR/AR/MR, and clarify the relationships among them [31], proposed the Reality-Virtuality (RV) Continuum, where AR and Augmented Virtuality (AV) lie on the two ends of the MR spectrum (see figure 10).

Figure 10: Reality-Virtuality (RV) Continuum [31]

On the very left end of the continuum lies the real environment, where users can only view the objects from the real world, either directly in person, or through an electronic display. AR concept is located at this end.

On the other side of the continuum, on the contrast, lies the virtual environment, where virtual objects generated by computer graphic simulations, are exclusively displayed in front of the users, either in a monitor-based or an immersive way. AV, i.e. VR, is presented here.

Between the extrema of the continuum lies the mixed environment, where both the virtual world and real-world objects are visible for users through one display. The MR is posed within this interval.

AR/MR system requirements

Three main features typically required for AR/MR systems are highlighted as follows [4]:

Firstly, it combines real and virtual objects. An AR/MR system should employ both the real objects observed in the environment and the virtual objects generated by the application. This brings about a link between the physical and augmented objects, which could be in any kind of form. Overlaying a text on an observed real object can be a type of such a link, whereas detecting the collision between the virtual objects and the real ones could be another.

Secondly, it runs interactively. An AR/MR system should also enable the users to interact with the virtual objects in the scene, as to measure up with the theme of "reality". How exactly users can interact varies from application to application. Going back to the previous examples, visualizing descriptive text on the objects in the real world is simple but still serves as a kind of interaction. On the other hand, the possibility of interaction can be, for example, to select, direct and place a virtual object in the real world as users intend to.

Thirdly, it runs in real-time. An AR/MR system should run all its processing and interactions in a real-time manner, which is a must to truly merge the computerized objects in the real-world scene. Accordingly, AR/MR applications need to be optimized to decrease the latency between user input and scene updates as much as possible. In the example mentioned before, as soon as its corresponding object appears in the scene and if it is the designed flow, the text overlays should become visible immediately at the right place. In the same way, a selectable and movable virtual object is expected to react instantly once the user takes corresponding actions with it, as a delay will possibly result in frustration and confusion for user experience.

AR/MR Displays

There are mainly two types of AR/MR displays: see-through (ST) displays and monitor/video-based displays [31].

ST displays allow the users to see the surroundings in the real world directly through the display's screen with top immersion and awareness. This is enabled by a transparent see-through screen and mirrors that overlay the digital content on the real environment. Most AR/MR HMD, including the Microsoft HoloLens used in this work, are ST displays.

Unlike ST displays dealing with the real-world scene, the monitor/video-based displays superimpose the computerized graphics on the live or stored videos. Mobile phones and tablets with AR applications running on them are typically categorized as such since the reality is formed by the video feed from the device camera.

The principles of these two types of displays are illustrated in figure 11.

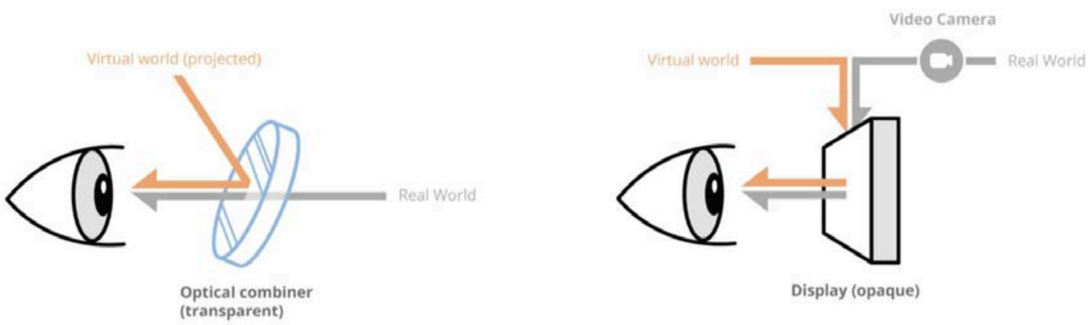


Figure 11: Principle of ST Displays (Left) and Monitor-based Displays (Right) [58]

AR/MR in industry 4.0

AR/MR technologies are expected to bring a disruptive transformation to the manufacturing and production industry. Industry 4.0 has already been witnessing plenty of applications of AR/MR, such as visualization of product 3D design, guidance on complicated assemblies tasks, the assistance of machine/process maintenance and of hands-on safety training, .etc.

The vacuum-pump manufacturer, Leybold, has utilized AR/MR technologies in the in-house sales activities and customer support process. For this, an AR/MR app has been developed both for HoloLens and iPad platforms. With the app, users can view and explore the information on the structure and different components of a vacuum pump, with no additional steps of disassemblies [33].

An "augmented smartFactory" app, through which operators can interact with a cyber-physical system, is developed utilizing AR/MR technologies. In detail, the App can visualize the critical information of the system into virtual objects and superimpose them directly into the real-world scene surrounding the user. Use cases of this app range from in-process monitoring, maintenance to intervene and servicing, base d on the visualization of the system's real-time status, interactive CAD or simulation models. Platform-independent software interface standards like OPC UA and web technologies are usually adopted to enable the information communication between the AR/MR interface and data gathered from CPS and the data integration within IIoT systems [10].

AR/MR systems developed and run on wearable devices such as smart glasses and head-mounted displays are stated to be advantageous, as the way of hands-free interaction enables users to operate the machines smoothly and effectively. When the AR/MR devices are utilized, no additional effort of checking manuals is required since all the related information can be overlayed directly in the real-world scene in front of users or even attached to the corresponding real-world objects. This will therefore reduce operational errors as comprehensive instructions are displayed right in the sight of users and easy to refer to.

According to a study by Boeing, AR/MR interfaces boost productivity and effectiveness in assembly tasks of wiring harness by 25% improvement. As reported by GE Healthcare, operators utilizing AR/MR interfaces at the warehouse could accomplish assembly tasks with a 46% higher speed. AR/MR interfaces are acknowledged to play a role in tackling the skilled worker shortages, as they could promote and ease the training process, empower the machine operators and free them for high-skill tasks, and equip their performance with high safety and productivity [1].

2.3 Industrial Communication and Distributed Systems

This section provides an overview of the industrial communication and networking related technologies, including OPC UA, GraphQL and REST. In the second part, distributed system technologies, i.e. parallel computing and multithreading in this context, are discussed as well.

2.3.1 OPC UA

Background of OPC UA

OPC Unified Architecture (OPC UA) is a widely-employed industrial standardized protocol for machine-to-machine communication. The standard is developed and maintained by OPC Foundation, while the specifications constituting the standard are constantly defined by end-users, developers and industrial partners together [34].

Prior to OPC UA, the market has already been impressed by its ancestor, OPC Classic with its availability for various Programmable Logic Controller (PLC) systems from different vendors. As it integrates various specific protocols for PLC into a standardized interface, it ensures coherency in the methods of accessing data, meaning that the accessing methods maintain regardless of the changes in resources or types of data. However, OPC Classic initially only works on the Windows operating system. Thus it is restricted to function as an interface standard between only Windows programs with other plant floor devices.

As a successor of OPC Classic, OPC UA was released in 2018, which enables the communication to be compatible with not only the Windows operating system but also other operating systems like Android, Mac OS, any distribution of Linux, and various embedded systems. Without the dependency on platforms, OPC UA allows data exchange and interface between clients and servers, along with clients and clients, which empowers even broader application fields in Industry 4.0, IIoT and M2M communication. Furthermore, its functionality is also enhanced in terms of efficiency and security, thanks to transporting data with structured information models and accessing OPC servers in a firewall-friendly manner.

OPC UA Features

OPC UA is characteristic of five features, which were set as goals in the developing process and all accomplished in the final released version [36].

The first feature is functional equivalence, meaning that all the functionalities and services supported by its ancestor OPC Classic should remain in the specifications of OPC UA, which include three parts. Data Access (DA) enables real-time bi-directional (both reading and writing) data sharing within industrial systems. Alarms & Events (AE) facilitates the data subscription, where clients can be notified when the node values change over certain thresholds or specific events occur. Historical Data Access (HDA) realizes access to historical data, allowing retrieving past values. Apart from the functionalities succeeded OPC Classic, there are also added ones, such as the AddressSpace, the definable methods running on servers, and the possibility of discovering OPC servers from local PC or networks,

Unlike OPC Classic constructed on COM, DCOM and OLE technologies that are developed by Microsoft thus only running for the Windows operating system, OPC UA is featured by platform independence. It is compatible with any hardware, such as computers, cloud servers, microcontrollers, and PLCs, as well as any operating system varying from Windows, Mac OS, Linux distributions, to Android and iOS.

The third feature is security, which is ensured by many design considerations of OPC UA. Data is transported through the firewall and many protocols can be utilized in this process. The OPC messages are encrypted typically with 128 or 256 bit and signed as well. In this way, messages stay the same when they are received as they are when they are sent. Security is even enhanced by sequencing packets, where message replay attacks are eliminated. Clients and servers in OPC UA are identified utilizing OpenSSL certification, where the communication access within each set of systems is definable. Besides, it is possible to determine or restrict access rights based on the users' authentication that can be requested by the application. Extended monitoring is also enabled as any actions on an OPC server are logged constantly.

OPC UA keeps an extensible architecture, meaning that new features and functionalities can always be introduced to the existing architecture in otorder to enhance the performance, whilst still being backward-compatible. This feature allows future-proofing the products and applications based on OPC UA.

The last but not least feature is comprehensive information modeling, where data is modeled into information, in the way that data is arranged in multi-level structure and in possession of object-oriented capabilities.

OPC UA Server Architecture

The architecture of an OPC UA server, together with the data flow among the server itself, between the server and clients or real objects, can be illustrated as figure 12.

The real objects represent either physical devices or software programs that are directly accessible by the server. The server application is an application that implements the functionalities of the server, which typically waits for requests from other applications then responds to them, thus providing a "service" upon their request. The AddressSpace is formed for a set of Nodes within the server, which ease the complicated data exchange process. AddressSpace can be further broken down into Views that define what Nodes are visible for clients. Subscription, as another kind of mechanism, can generate Monitored Items, which is used to keep track of the Node values, when it changes or reach a pre-defined threshold, clients will get notifications accordingly [35].

2.3.2 REST and GraphQL

REST Overview

REST (REpresentational State Transfer) is an architecture style for distributed hypermedia systems [45]. It was firstly introduced by Roy Fielding in 2000 in his dissertation, where, like other architectural styles, REST is presented together with its six guiding constraints or design criteria, i.e. client-server, stateless, cacheable,

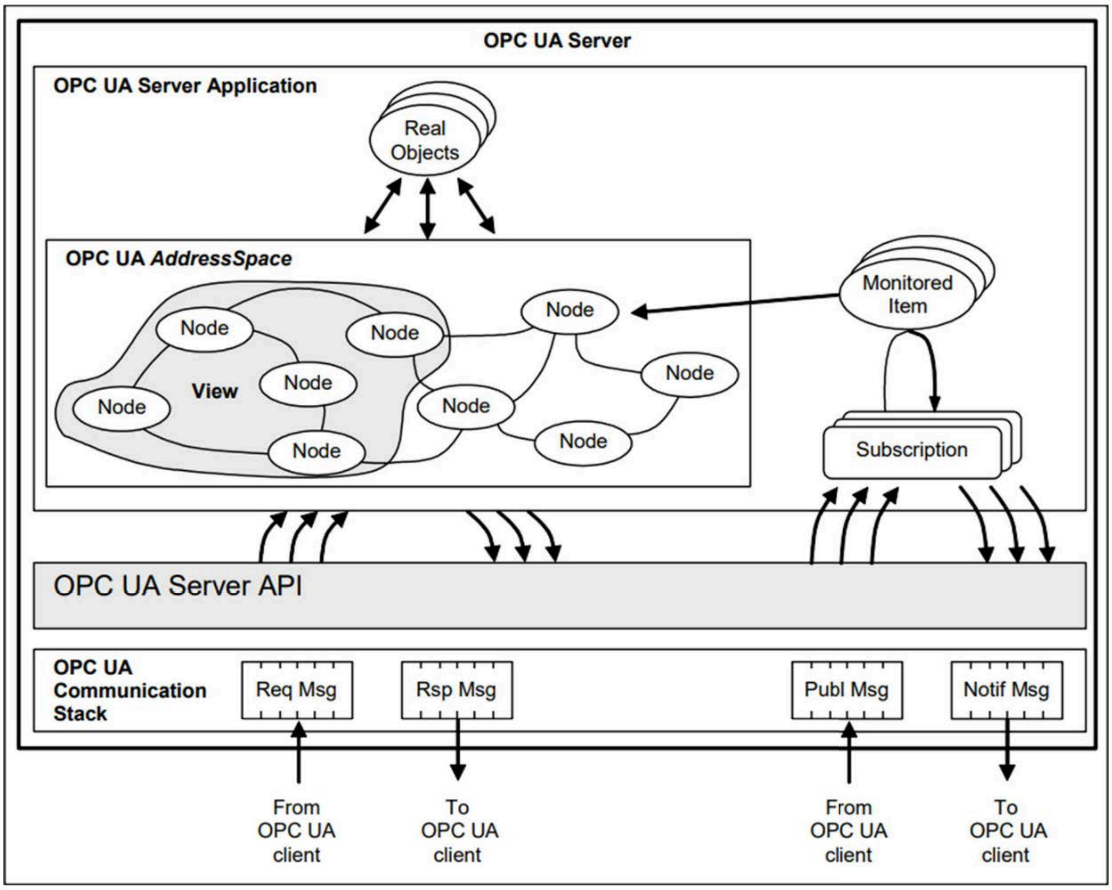


Figure 12: OPC UA Server Architecture [35]

uniform interface, layered system and code on demand (optional). A software or interface can be referred to as RESTful, if the six guiding constraints are fulfilled [8].

In REST, the abstraction of information is called resource. A RESTful system can be implemented utilizing Resource-Oriented Architecture (ROA) [45], where every resource carries an address, either explicitly or implicitly, in order to be identified and accessed during interaction among different components and objects. Furthermore, the resource methods, which can be used to perform the transitions and interactions of resources, should be uniform interfaces. For instance, the interface can utilize HTTP and related methods like GET, POST, PUT, DELETE on the web.

Although one of the intentions of REST is to standardize the web (internet), the REST architectural style is, however, not bound to any protocol preference [13], let alone the common misunderstanding that REST and HEEP are the same. Yet, the REST architectural style is the most popular way to implement web applications. RESTful applications are enabled by their advantageous design principles to be lightweight, uncomplicated, fast and highly scalable.

GraphQL Overview

GraphQL is a query language and specification for APIs and a runtime for performing queries with existing data on the server. It has been firstly developed and adopted only internally at Facebook since 2012. In 2015, the open-source version was publicly released [11].

GraphQL does not specify any instruction on certain programming languages or platforms. Therefore, it eases the cross-platform project and mobile developments with the availability on a wide scale. Moreover, GraphQL APIs can significantly diminish the amount of data transferred and thus the network traffic by fetching all the data needed in a single request. An example of the data transferring flow is shown in 13. Thanks to its user-centered design principles that focus on developers' experience, and its micro-services architecture that emphasizes the speed to market, GraphQL enables higher productivity, efficiency and flexibility.

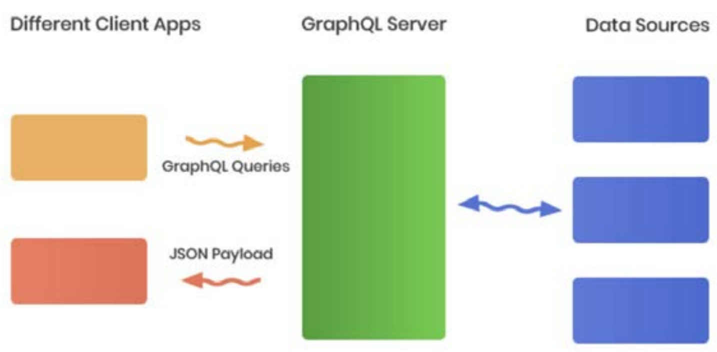


Figure 13: Data Transferring in GraphQL [21]

GraphQL has become increasingly prevalent in the past years. Nowadays, it is adopted at the production level by numerous organizations of different volumes, including Facebook, GitHub, PayPal, the New York Times, and more.

REST and GraphQL Comparison

The purpose of either REST or GraphQL is to fetch data through APIs, where, additionally, HTTP protocols are typically utilized in both cases.

Yet, REST APIs commonly adopt HTTP methods such as GET, POST, PUT, DELETE to implement the operations like Create, Read, Update, Delete (CRUD), whilst GraphQL adopts methods specific to API and supports queries with the GET and POST requests.

GraphQL is also distinct from REST in the way that with one GraphQL query multiple resources can be accessed, while, on the other hand, REST strictly follow the ROA as mentioned before, meaning that each resource corresponds with its very URL and can be interacted with only at one time through one single query (see figure 14).

Figure 14: REST vs. GraphQL (Multidots) [21]

2.3.3 Parallel Computing and Multi-threading

Parallel Computing in .NET

Multiple CPU cores are usually available in many computers, workstations, mobile devices, and overhead displays, for instance, the HoloLens 1 used in this work has four logical CPU cores. Taking advantage of this, it is possible to distribute computation work across multiple processors by paralleling the program [25].

In order to parallel the code, previously it was necessary to manipulate the threads and locks at a low level. Now thanks to the .NET framework, which provides runtime, class library types, together with the diagnostic tools, parallel programming gets simplified and more accessible.

Furthermore, many features in .NET Framework empower the parallel development in an intuitive manner with improved efficiency and scalability, as it is not required to manually handle threads or the thread pool.

The architecture of the parallel computing in the .NET framework is depicted in the figure 15 to provide a high-level overview.

Multithreading Overview

With a single-threaded system, where an instruction inputs then the corresponding result outputs one after another, the entire time cost to carry out the whole program is therefore determined solely by the workload assigned to the CPU.

Multithreading is a way of computing and programming, which utilizes a CPU's capability to provide multi-threads of processing and execution across several cores simultaneously.

The "main thread", which by default runs once the program begins, can generate new threads to process tasks. The generated threads share the resources of a single core

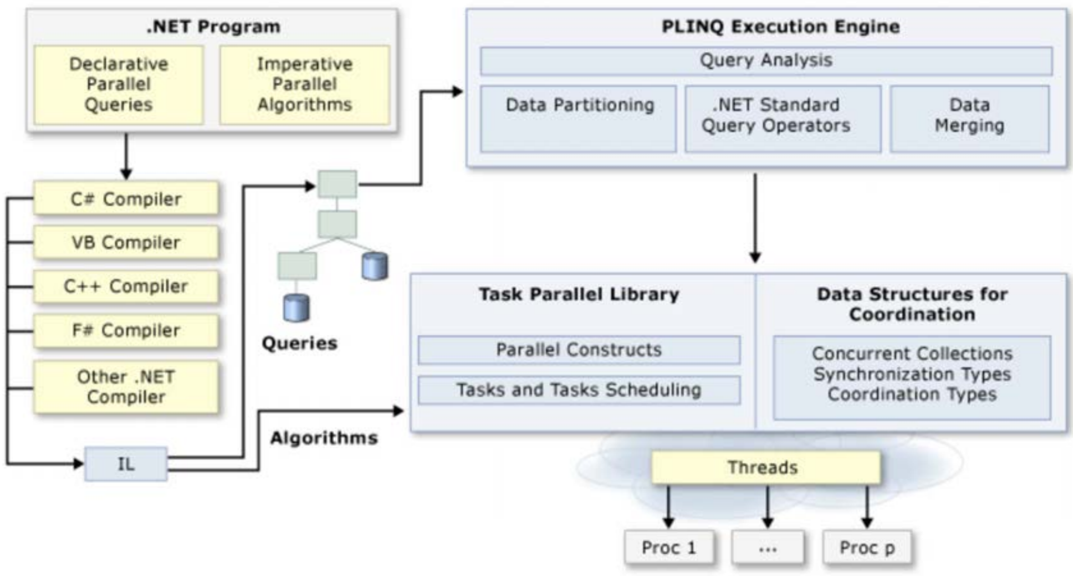


Figure 15: Parallel Programming Architecture in the .NET Framework [25]

or multiple cores, run in parallel manner, and typically communicate and synchronize the outputs with the main thread once one is done.

The multithreading approach is worth applying only in the case where only a few tasks are in execution for a sufficiently long time compared to the entire program running time. For instance, the usage of multithreading in game development needs paying special attention, since a game program often consists of numerous short instructions to execute, which could potentially overload the processing capability of the CPU and operating system by too many threads with short lifetime [52]. An alternative way to solve this issue can be to have several threads in a thread pool. However, possibilities could be that quite many threads are still in active status simultaneously.

As a result of overloading the CPU core with more threads than the CPU resources could support, context switching will occur. During a typical context switching process, the state of a thread is saved halfway of its execution, then the program switches to another thread, then later on it reconstructs and switches back to the previous thread for continued execution. Such a process consumes a large number of resources, which should be prevented from happening.

2.4 Tracking and Registration

This section provides an overview of tracking, calibration and registration techniques in AR/MR applications, with highlighting the common approaches, practices, as well as the principles and usage of data flow networks (DFN) and graphs (DFG), spatial relationship graphs (SRG) and patterns (SRP).

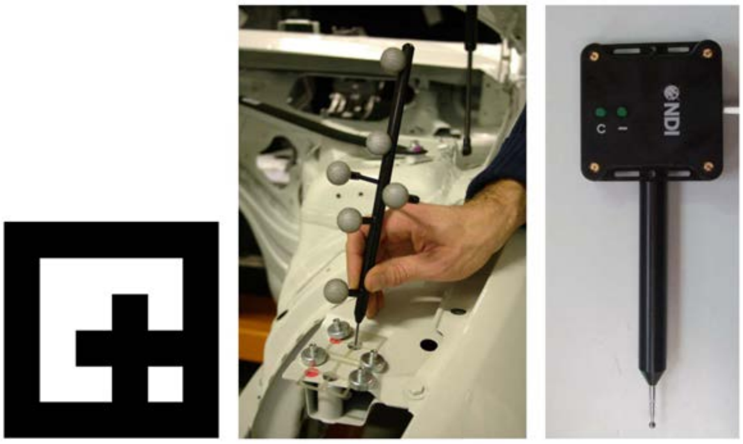
2.4.1 Tracking Systems

Tracking Overview

Tracking represents a major challenge in the context of MR/AR, where the pose of an object including its position and orientation in relations with a certain coordinate system is continuously being detected in a real-time manner.

Tracking in typical Industrial AR/MR applications is supposed to fulfill at least one of the following requirements [18]:

- To register the computer-generated 3D content that could be moved around in the physical real-world space;
- To register the mobile displays like handheld devices or head-mounted displays with the virtual world;
- To register the mobile physical objects such as markers and pointing devices (see figure 16) with the virtual world.



(a) Square marker (b) Pointing device with passive retro-reflective fiducials (c) Pointing device with active IR LEDs

Figure 16: Square Marker and Different Pointing Devices [18]

Different sensors can be adopted to enable tracking in AR/MR application, where, for instance, optical cameras are often used to fetch data of high-quality. Meanwhile, a variety of approaches can be employed, which share the common feature that 3D information is recovered with certain 2D points in the image through geometric relations.

Tracking Approaches

One of the most common tracking systems is based on the marker, which is an object or image with a distinctive structure ("fiducials") so that it can be easily segmented out of the whole scene. In Industrial applications, the typical markers are either the passive type that is built from retro-reflective fiducials or the active type that is made of infrared LEDs, which can all be easily tracked by infrared cameras. Besides, tracking also works well with printed square markers with a particular shape as content (see figure 16 a).

Pointing devices are also often adopted in tracking applications. They can be either equipped with a pike tip (see figure 16 b) or with a spherical tip (see figure 16 c), which is more common, especially in the domain requiring high accuracy. The measurement tasks typically include those on drilling and clearances, which can be carried out seamlessly with high precision through multiple measurements and directly touching the object's surface with the sphere tip. Until a few years ago, most AR/MR applications fell under the category of marker-based AR, but today more and more marker-less AR solutions have evolved into various SDKs that produce the same effect without the need for specialized equipment. Instead of relying on fiducials, marker-less approaches utilizing extracted distinctive point or line features. The detection, recognition and classification of those features are enabled by either prior learning or utilizing digital information out of the scene.

Global tracking approaches like utilizing GPS or compasses do not often lead to measurements of precision that is high enough for industrial use cases, but it often still makes sense to adopt them to initial other trackers. However, due to deviations caused by metal objects, the utilization of compasses is usually not that meaningful in the industrial environment.

With the Inertial Measurement Unit (IMU) like gyroscopes and accelerometers, motion data can be measured and utilized for tracking. However, this approach suffers from the accumulated drift of the measurement, whose amount highly depends on the hardware dimension. As for the AR/MR mobile devices usually of small size, large drift can occur, resulting in invalid measurements.

Other possible methods for tracking include online metrologic tracking, laser tracking, optical or acoustic tracking [18].

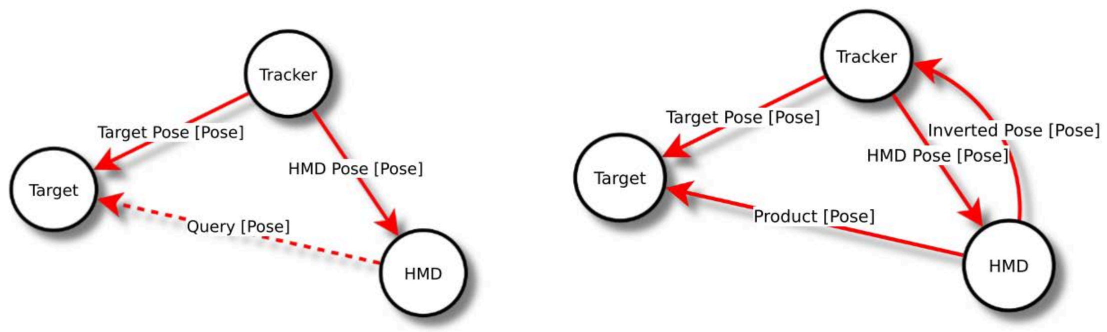
2.4.2 Spatial Relationship Graphs (SRG) and Patterns (SRP)

Spatial Relationship Graphs (SRG)

Spatial Relationship Graphs (SRGs) represent directed cyclic graphs, which specify the static and dynamic spatial properties of real or virtual objects and thus depict a high-level infrastructure of the tracking environment.

In SRGs, the nodes illustrate the local coordinate frame of digital or physical content (i.e. objects, scene) as well as sensors. The directed edge from node A to node B depicts the spatial transformation H_A^B from the coordinate frame A to frame B, in other words, the six-DoF pose information (position and orientation) of frame B with regard to frame A. Similarly, $p_{A_i} = H_A^B p_{B_i}$ describes the transformation of the point p from frame B into frame A [43].

Figure 17 (a) illustrates an example of a “base SRG”. The node titled “tracker” is



(a) Base SRG (b) Full SRG

Figure 17: Examples of Base SRG and Full SRG [43]

a sensor that tracks the poses of the objects, represented by the two nodes below, one titled “HMD” that refers to a head-mounted display, one titled “target” that represents either a virtual or real target object. The directed solid edges illustrate the spatial relationship between “Tracker” and “HMD”, as well as the one between “tracker” and “Target”, which are measured and updated constantly and dynamically. The dashed edge between “Target” and “HMD” refers to the query to fetch the spatial relationship that can not be directly measured but derived from the tracking data indicated by the solid edges.

Figure 17 (b) depicts the corresponding “full SRG”, which is free of any dashed edge for the query, but instead with additional solid edges indicating the mathematical operations behind the scene. In this example, two new solid edges, one indicating the inversion on the spatial relationship between “Tracker“ and “HMD”, the other indicating the concatenation of the inverted “Tracker” pose relative to “HMD” and the “Target” pose relative to “Tracker”.

Data Flow Networks (DFN) and Graphs (DFG)

The SRG illustrates the data flow of a tracking system in a descriptive and abstract manner, however, it does not include any operational specifications that are critical in real applications. Therefore, we need to convert SRG into a Data Flow Networks (DFN), which is composed of the computational units operating on tracking data.

The Data Flow Graphs (DFG) is a directed graph with its nodes specifying the data flow components that are instantiated in DFN. In other words, DFN is an instance of DFG. In DFGs, the data flow component possesses the input ports that correspond to dashed edges in SRGs, and the output ports that correspond to solid edges in SRGs. This means that input ports employ tracking data requested by the component, while output ports provide the computation results on the tracking data. The edges in DFG depicts the tracking data flow from the output port of one data flow component to the input port of the other. In DFGs, the sources correspond to the tracking devices, as they are the source of the tracking data, while the links represent the interfaces to applications or other DFGs [43].

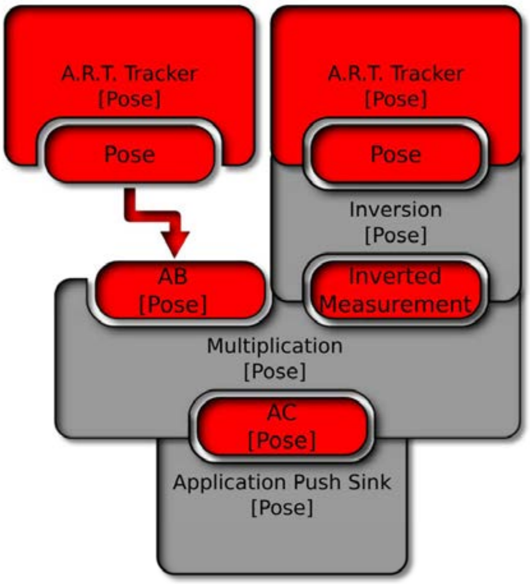


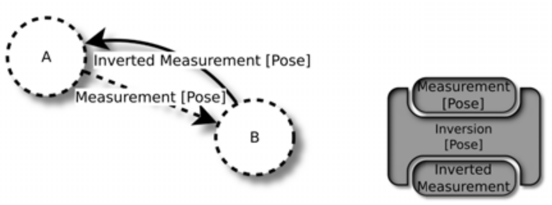
Figure 18: A DFG Corresponding to SRG Example in Figure 17 (b) [43]

Figure 18 illustrates the DFG that corresponds to the SRG depicted in figure 17 (b). In this exemplary DFG, the tracked pose data of the HMD is firstly operated through inversion, then the inverted measurement is concatenated together with the tracked pose data of the target to compute the dashed edge in figure 17 (a). The red color implies that the sensors are constantly driving the update of the measurements of the tracking data and the computation results [39].

Spatial Relationship Patterns (SRP) Overview

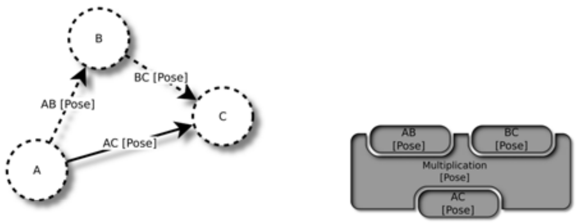
A Spatial Relationship Pattern (SRP) represents a template SRG, which is in correspondence to a data flow component in a DFG. Each SRP is linked with a unique pattern ID out of the whole DFG, which is required while interfacing external data flow components [41].

As described in the previous section, the data flow component receives tracking data through input ports, then sends out the computation results through output ports. Figure 19 illustrates an example of the inversion SRP out of the SRG in figure 19 (a) and its corresponding data flow component out of the DFG in figure 19 (b). Analogously, figure 20 depicts the multiplication SRP and its corresponding data flow components. These two SRP can both be categorized as “full patterns”, whose corresponding data flow component has neither empty inputs nor empty output, and thus generating new information out of existing data [42].



(a) Full pattern (b) Data flow component

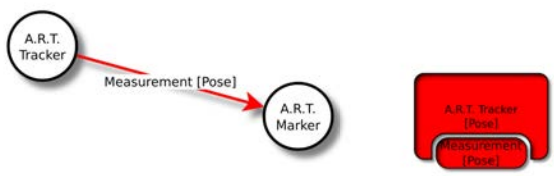
Figure 19: Full Pattern SRP - Inversion (a) and its Corresponding Data Flow Component (b) [43]



(a) Full pattern (b) Data flow component

Figure 20: Full Pattern SRP - Multiplication (a) and its Corresponding Data Flow Component (b) [43]

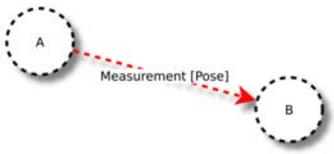
Another category of SRP is the base pattern, whose corresponding data flow component has empty input but non-empty output. This indicates the tracking data is directly placed into the DFN. An example of the base pattern and its corresponding data flow components that typically represents a tracking device is shown in figure 21.



(a) Base pattern (b) Data flow source component

Figure 21: Base Pattern SRP (a) and its Corresponding Data Flow Component (b) [43]

In contrast to the base pattern, the query pattern depicted in figure 22 is a pattern with non-empty input but empty output. It typically represents the data sinks that transfer tracking data from the current DFN to the application interface or other DFNs.



(a) Query pattern

(b) Data flow sink component

Figure 22: Query Pattern SRP (a) and its Corresponding Data Flow Component (b) [43]

2.4.3 Calibration and Registration

Concept of Calibration and Registration

The concept of calibration represents the estimation of the parameters describing the actual behaviour of a physical object [18]. In the context of tracking for AR/MR applications, the term registration is used to specify the estimated static spatial relationship transformations in SRGs conditioned by certain boundaries. For instance, a node B is registered with regards to node A when the static edge between A and B is acknowledged.

Registration aims at determining static offsets between the coordinate systems of markers, sensors and other objects. Common registration algorithms include Hand-Eye Calibration, SPAAM Calibration, Tip Calibration (despite they confusedly have “calibration” in the name), 2D-3D Pose Estimation or 3D-3D Pose Estimation [40].

The data flow of the registration process typically has the same structure as the one in other tracking applications, which can be depicted completely by DFGs and SRGs.

While sensors are tracking the objects in the environment during the registration process, AR/MR developers usually need to move the objects according to the selected registration method. The tracking data coming from sensors are stored and used to compute and determine the static transformation information.

Calibration / Registration Patterns

The calibration / registration patterns can be categorized as follows [43].

Figure 23 (a) illustrates the Absolute Orientation pattern, which utilizes minimum three 3D pose measurements of the same feature to generate the estimation of the pose transformation.

Figure 23 (b) depicts the Hand-Eye Calibration pattern, which utilizes minimum two pose measurements of the robot relative to the hand and of the hand relative to the eye in order to estimate the hand-eye pose transformation. This algorithm is originally applied in the robotics field for estimating the offset of a camera mounted

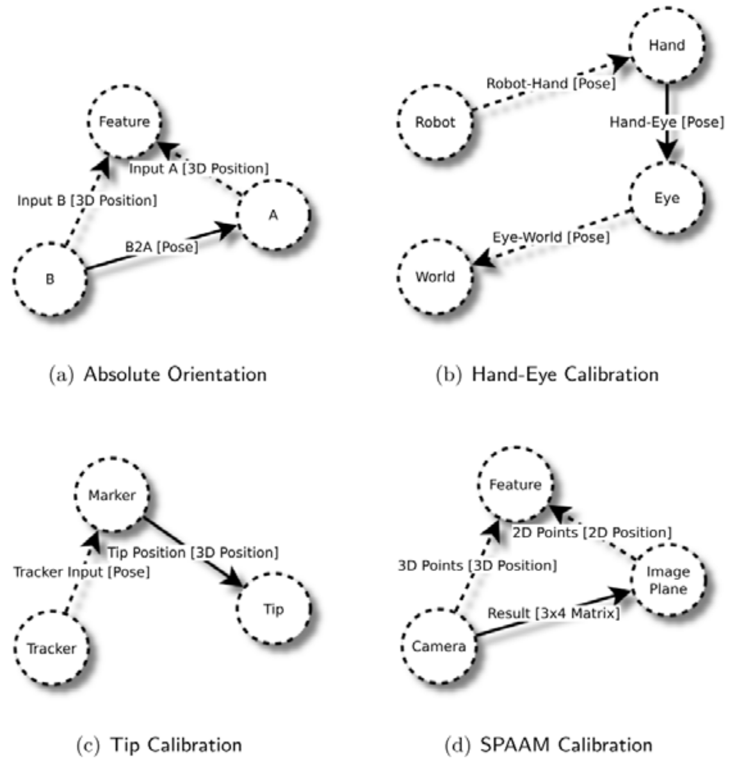


Figure 23: Common Registration Categories [43]

on a robot arm. Developers can therefore perceive or extract the pose of the robot arm from the robot kinematics and the static reference target out of the camera view. [50]

Figure 23 (c) presents the Tip Calibration pattern, which makes estimations on the 3D pose offset of a tip relative to the marker typically attached to the end of the device. Several measurements should be conducted with different poses of the marker while the tip stays still.

Figure 23 (d) shows the SPAAM Calibration pattern, which generates estimation on a 11-DoF projection matrix out of at least six 2D and 3D position measurements. The registration is enabled by manually adjusting the displayed 2D points to align with a 3D point based on the word coordinates from different views. With tracking data of the marker's pose relative to the world coordinates, the 3D point's pose can then be converted to the marker's coordinate system. Since the marker is fixed to the display, the display can be moved during the calibration process.

A variety of alternative patterns are also available to solve the registration task.

3 Hardware Setup

This chapter introduces the hardware adopted in this work, i.e. the Microsoft HoloLens (1st Generation), the smart crane "Ilmatar" and its operating environment AIIC, with highlighting their supportive features in the context.

3.1 Microsoft HoloLens (1st Generation)

This section will provide an overview of HoloLens, as well as its specifications and capabilities.

3.1.1 HoloLens Overview

The HoloLens, developed and manufactured by Microsoft, is the first head-mounted device running the Windows Mixed Reality (WMR) platform under the Windows 10 computer operating system, empowered with the holographic immersive display, advanced optics and sensor technologies [27].

WMR is a Microsoft platform constructed with Windows 10 API, allowing apps to blend the real spatial environment with computer-generated virtual content on either MR immersive headsets or holographic devices such as HoloLens. The users can therefore interact with the digital content consisting of light and sound that appears like parts of their surroundings, as if with objects from the physical world. Those digital content, which is called Hologram, can respond to the users' gaze, gesture, and voice commands, and can also interact with the physical surface in the real world.

The pre-production version and the 1st generation of HoloLens were released in 2016, initially for developers to experiment and deploy both MR and AR applications. Then in the year 2019, the HoloLens 2 was launched with enhanced features.

3.1.2 Device Specifications

Display

The display of HoloLens, which is also termed as the holographic frame, is the fundamental and most crucial component. Empowered by the see-through holographic lenses with high holographic resolution and density, as well as the eye-based rendering and automatic pupillary distance calibration, users can have the immersive experience with virtual content enhancing the physical world surrounding them. The display specifications are shown in figure 24. '

Sensors

Various sensors are embedded in HoloLens to enable the perception of the environment, the placement of Holograms in the surroundings, as well as the understanding of user interaction. Figure 25 illustrates these sensors, which include one IMU (with accelerometer, gyroscope and a magnetometer) for helping to track the movements

Display

Optics	See-through holographic lenses (waveguides)
Holographic resolution	2 HD 16:9 light engines producing 2.3M total light points
Holographic density	>2.5k radiants (light points per radian)
Eye-based rendering	Automatic pupillary distance calibration

Figure 24: HoloLens Display Specifications [27]

of the user, four "environment understanding" cameras (two on each side), one energy-efficient depth camera, one photo/video camera, one ambient light sensor, as well as a four-microphone array.

Sensors

- 1 inertial measurement unit (IMU)
- 4 environment understanding cameras
- 1 depth camera
- 1 2MP photo / HD video camera
- Mixed reality capture
- 4 microphones
- 1 ambient light sensor

Figure 25: HoloLens Sensor Specifications [27]

Processors, Power and Memory

Due to the heavy workload to handle the data for sensors and holographic display, HoloLens is equipped with powerful processors, including one Intel 32-bit architecture processor and one custom-built Holographic Processing Unit (HPU 1.0). The battery life of HoloLens supports 2-3 hours of active use and up to 2 weeks of standby time. It is fully functional while charging and featured by passively cooling (without fans). Furthermore, 64GB Flash and 2GB RAM are available (see figure 26).

Figure 26: Specifications of HoloLens Processors, Memory and Power [27]

Input, Output, and Connectivity

Regarding Input and output, HoloLens has built-in speakers and an audio 3.5mm jack. There are buttons for power and to adjust volume and brightness up/down. The LEDs indicate its battery status.

Networking technologies like Wi-Fi 802.11ac, Bluetooth 4.1 LE, and wired connectivity using micro USB 2.0 enable the connectivity of HoloLens (see figure 27).

3.1.3 Device Capabilities

HoloLens 1 is capable of understanding the user actions through three ways, i.e. gaze tracking, gesture input, and voice command.

Figure 27: Specifications of HoloLens Input, Output, and Connectivity [27]

Gaze Tracking

HoloLens' capability of gaze tracking enables users to give the input based on which direction they are looking at. The process of navigating the virtual cursor in the hologram-enhanced physical world with the head (see figure 28) is similar to how the user can do with a mouse across a screen. The gaze input empowers the users to interact with the environment firstly in a head-free manner, where they typically gaze to select a certain digital content, then perform either gesture or voice input to take actions on the chosen object [28]. While HoloLens 2 supports the advanced eye-gaze input, HoloLens 1 is only associated with the head gaze input, and typically combined with the commit methods described above to perform a complete interactive action.

Gestures Input

Gesture input in HoloLens supports interactions with bare hands without requiring users to hold controllers like in other head-mounted displays (see figure 29).

HoloLens is equipped with the camera sensors that can perceive a frame of a few feet length between two sides of the user. Hand gestures are therefore enabled only if the hands are shown inline with the sight of the HoloLens and sensed by the device camera. As a result, if it is adopted too often, the physical fatigue, i.e. the gorilla arm, could possibly occur [30].

There are two ways of gesture inputs that are supported by HoloLens 1, i.e. bloom

Figure 28: HoloLens Gaze Input [24]

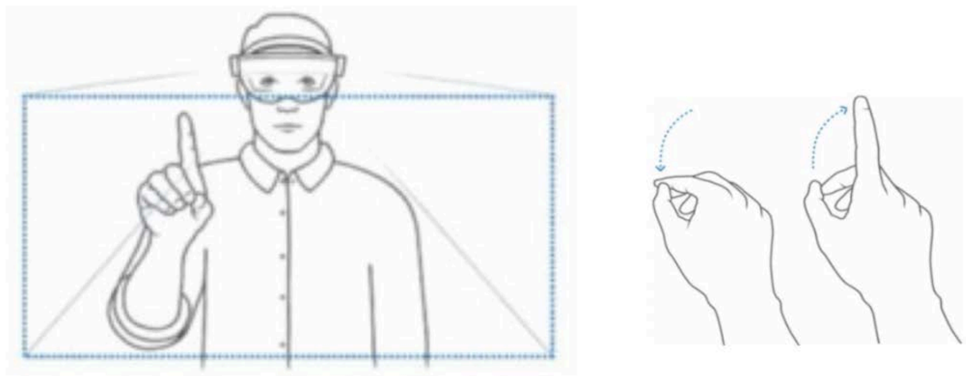


Figure 29: HoloLens Gesture Input [24]

and air tap. While the bloom gesture typically servers to adjust (open/hide) the main menu, navigation bar and apps, as well as to re-orientate, the air tap gesture, along with the gaze, enables users to select holograms and gaze/dwell buttons within the scene [28].

To use the bloom gesture, the user needs to (see figure 30):

- Firstly hold the hand straight out in front with the palm facing up and fingertips all together;
- Then, open the hand with fingers stretching.

To use the air tap gesture, the user needs to: (see figure 31)

- Firstly, gaze at hologram to select;
- Secondly, hold the hand out with a loose fist and the index finger pointing straight up;
- Lastly, press the index finger down then quickly retrieve the raising up position.

Besides, there is the combined gesture of air tap and hold, where at the end of the air tap gesture, users maintain the downward position of the index finger while moving the arm, instead of raising it back up again, in order to drag the selected/activated

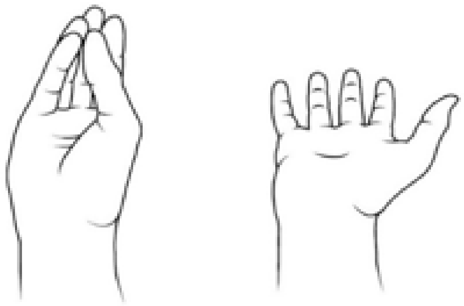


Figure 30: Bloom Gesture [30]

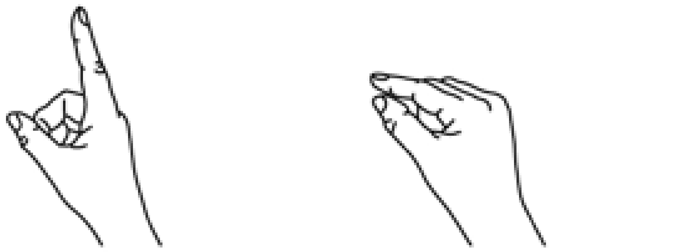


Figure 31: Air Tap Gesture [30]

hologram around. It worth taking into consideration in designing that users are likely to ease off the gesture during arm movement, which might cause unexpected results.

Voice Support

Voice input enables the user to interact with holograms through speech command in a hand-free manner (see figure 32). It provides a natural and intuitive communication way. The common usages of voice support are adjustments of holograms, navigation through a large menu, or single-message instruction [24].

3.2 Smart Crane "Ilmatar" and AIIC

This section will provide an overview of the smart crane "Ilmatar" and its operating environment AIIC, as well as the movement subsystems and smart features of the crane.

Figure 32: HoloLens Voice Input [24]

3.2.1 Crane Overview

The overhead smart crane adopted in this work, named "Ilmatar" (see figure 33, figure 34), is manufactured by the Finnish company Konecranes, and donated to the Aalto Industrial Internet Campus (AIIC) of Aalto University, functioning as a digital twin based IIoT platform for students and researchers to conduct experiments on Industry 4.0 related use cases.

Figure 33: Crane "Ilmatar" look 1 [16]

The"Ilmatar" crane is empowered by various smart features. For instance, sway control could minimize the sway caused by acceleration and deceleration; Inching provides a way for accurate load positioning; Micro speed makes the load control more precise; Target Positioning allows work cycles to be carried out using only two buttons on the radio controller. Those features can be enabled through the crane's control panel, which potentially leads to improved efficiency, safety and productivity.

Figure 34: Crane “Ilmatar” look 2 [16]

3.2.2 Crane Movement Subsystems

The “Ilmatar” crane is composed of three subsystems of different translational movements, i.e. the hoist for moving up and down, the trolley and the bridge for moving backward and forward of different directions (see figure 35, 36). The whole movement system enables the crane to complete the task of lifting its load within its capacity from one spot to the other among the moving range and within the speed limit of each subsystem.

3.2.3 Crane’s Operation Environment: AIIC

Figure 37 and figure 38 are the screenshots of the 3D model constructed with the Visual Component software by Jouni Hannonen (as a separate work that has not been published), which illustrate how the crane and its operation environment AIIC looks like.

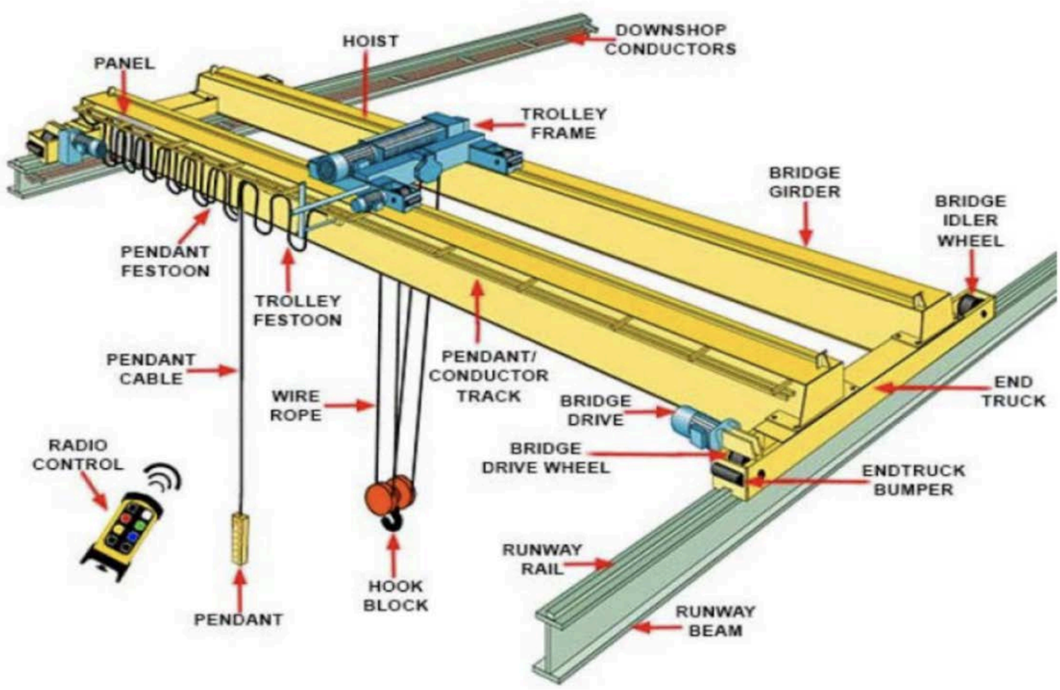


Figure 35: Crane “Ilmatar” Components [17]

Subsystem	Feature	Value
Hoist	Lifting Height	3 meters
	Lifting Speed	8 meters/ minute
	Lifting Capacity	3200 kg
Trolley	Moving Range	9 meters
	Moving Speed	20 meters/ minute stepless
Bridge	Moving Range	9 meters
	Moving Speed	20 meters/ minute stepless

Figure 36: Crane “Ilmatar” Subsystem Features [17]

Figure 37: Crane “Ilmatar” and AIIC Model - Look 1

Figure 38: Crane “Ilmatar” and AIIC Model - Look 2

4 Software Setup

This chapter introduces the software adopted in this work, which is explained in two sections. One is the software available in the crane platform, including the general software system of the "Ilmatar" crane, with its modules of OPC UA and GraphQL that support connectivity being highlighted. The other is the software and SDKs used for MR application development, including the game engine Unity with MRTK as the fundamental developing platform, the Vuforia SDK for space registration, RestSharp for communication with crane GraphQL, and the Task Parallel Library (TPL) in .NET for multithreading.

4.1 Software System and Connectivity of the Crane

4.1.1 Crane Software System Overview

The "Ilmatar" crane is empowered by the software system composed of PLC, OPC UA server, remote controller system software, UI server, and remote monitoring system, which are of significance for research, education and innovation. Figure 39 provides an overview of the software platform environment, with the software, the purpose and location of each listed.

Software System	Purpose	Location
PLC	Handle crane's sensor data and control system.	Crane
OPC UA Server	Provides external access to crane data and controls (hoist, bridge, trolley)	Crane
Remote Controller System Software	Enables external control of the crane.	External Equipment
UI Server	Provides external access to the web UI.	Crane
Remote Monitoring System	Provides external access for monitoring.	Crane

Figure 39: Crane "Ilmatar" Software System Overview [17]

4.1.2 Crane OPC UA Interface and Data Access

The "Ilmatar" crane interface utilizes OPC UA for communication and control, empowered by various security features to handle the connection break or errors caused by external systems. The necessity of these security features lies on that, on one hand, the connection with OPC UA is enabled through a Wi-Fi network, which can not guarantee to be stable all the time; on the other hand, quick programming and deployment without thorough checking and testing often result in bugs in the program.

In order to control the crane, users firstly need to be identified their personal access code, with the system storing the date and time when an access code has been used. Then the watchdog value of the crane needs to be modified constantly, with a message activated in the maintenance interface to indicate that the watchdog is running. This means that the crane will stop accepting commands and the control will be interrupted once the application collapses or the connection breaks down.

In addition, only if a certain functional button (see figure 40) on the side of the crane remote controller is pressed continuously, that the crane can accept any control command from an external application. This means that the crane movement can be terminated immediately by releasing the controller button, in case of any emergency. Furthermore, the crane can be turned down instantly also through the emergency stop knob of the remote controller.

Figure 40: Crane "Ilmatar" Radio Controller with controller button highlighted [19]

All these safety measurements ensure a secure on-the-fly development and testing loop for applications communicating with crane OPC UA.

The crane OPC UA server has multiple variables available for writing and reading. The control interface has different data structures for the control signals for manual control (see figure 41) and target positioning control (see figure 42), as well as status signals (see figure 43).

As for reading, there are the radio controller signals that represent the status of the radio controller selections (see figure 44), the machinery signals for hoist (see figure 45), trolley, bridge and target positioning separately with different data structures, as well as the common crane signals that related to the whole system.

Variable name (prepend SCF.PLC.DX_Custom_V.Controls.)	Data type	Description
Watchdog	Int (16bit)	Has to change constantly
AccessCode	DInt (32bit)	Personal Access code
Hoist.Up	Bool	1 = Up / Position increases
Hoist.Down	Bool	1 = Down
Hoist.Speed	Float (32bit)	0 - 100 % / Maximum speed is limited by the PLC SW
Trolley.Forward	Bool	1 = Forward / Position increases
Trolley.Backward	Bool	1 = Backward
Trolley.Speed	Float (32bit)	0 - 100 % / Maximum speed is limited by the PLC SW
Bridge.Forward	Bool	1 = Forward / Position increases
Bridge.Backward	Bool	1 = Backward
Bridge.Speed	Float (32bit)	0 - 100 % / Maximum speed is limited by the PLC SW

Figure 41: Control Variables - Manual Controls [20]

Variable name (prepend SCF.PLC.DX_Custom_V.Controls.TargetPositioning)	Data type	Description
SelectionInUse	Bool	1 = Target Selection from OPC UA
DriveToTarget	Bool	1 = Drive to Target
DriveToHome	Bool	1 = Drive to Home Target
Target	Int (16bit)	Target Positioning target number, [1..200], 0= No target selected
Home	Int (16bit)	Target Positioning home target number, [1..200], 0= No target selected

Figure 42: Control Variables - Target Positioning Controls [20]

There is also an OPC UA python library available on the crane platform [2], which was developed based on the "Free OPC-UA Library". It contains functions to fetch crane position and move the crane through its OPC UA interface.

4.1.3 Crane OPC UA - GraphQL Wrapper

Although OPC UA is a standardized protocol commonly adopted in industrial applications, software engineers, especially web developers, are typically not familiar with OPC UA, or find it complicated to use. In order to enhance and ease the software development process for the crane, a GraphQL wrapper for OPC UA [15] has been developed with the help of the OPC UA python library [2] and available in the crane platform (see the project in GitHub [3]). The wrapper enables the clients to access the crane OPC UA servers through a GraphQL API with the information model node data available in the interface. An example of the communication flow between the OPC UA server, GraphQL interface and client is illustrated in figure 46.

In detail, the wrapper could convert the easy-to-use GraphQL queries from client

Variable name (prepend SCF.PLC.DX_Custom_V.Status.)	Data Type	Description
ReadingControls	Bool	1 = Controls are read from OPC interface
WatchDogFault	Bool	1 = Watchdog fault active / Watchdog value is not changing

Figure 43: Control Variables - Control Status [20]

Variable name (prepend SCF.PLC. DX_Custom_V. RadioSelection)	Data type	Description
Inching	Bool	1 = Inching is activated from radio controller
MicroSpeed	Bool	1 = MicroSpeed is activated from radio controller
RopeAngleFeaturesBypass	Bool	1 = Rope Angle Feature Bypass is activated from radio controller
SwayControl	Bool	1 = Sway Control is activated from radio controller
SwayControl_SlingLength_mm	DInt (32bit)	Sway Control Sling length selection in millimeters.

Figure 44: Variables (Read) - Radio Selection Variables [19]

Variable name (prepend SCF.PLC.DX_Custom_V.Status.Hoist)	Data type	Description
Diagnostics.Ok	Bool	1 = Machinery OK
Diagnostics.Event_Active	Bool	1 = At least one event is active
Diagnostics.Alarm_Active	Bool	1 = At least one alarm is active
Diagnostics.Fault_Active	Bool	1 = At least one fault is active
Diagnostics.Bypass_Active	Bool	1 = At least one bypass is active
ControlSignals.Direction1_Request	Bool	1 = Driving command on from active control place (Up / Forward)
ControlSignals.Direction2_Request	Bool	1 = Driving command on from active control place (Down / Backward)
ControlSignals.Speed_Request	Float (32bit)	Speed Reference in % from active control place
ControlSignals.SpeedFeedback	Float (32bit)	Current speed in %
ControlSignals.SpeedFeedback_mmin	Float (32bit)	Current speed in m/min
ControlSignals.MotorTorque	Float (32bit)	Motor Torque %, Positive when motoring and negative when generating
Position.Calibrated	Bool	1 = Position is calibrated
Position.Position_m	Float (32bit)	Current Position in meters
Position.Position_mm	DInt (32bit)	Current Position in millimeters
Position_Raw	DInt (32bit)	Raw position measurement in millimeters
Position.UpStop_m	Float (32bit)	Up stop limit position in meters
Load.Load_t	Float (32bit)	Current gross load in tons
Load.TaredLoad_t	Float (32bit)	Current tared load in tons
Load.GrossLoad_Percentage	Float (32bit)	Current gross load in %
Load.TaredLoad_Percentage	Float (32bit)	Current tared load in %

Figure 45: Variables (Read) - Hoist Machinery Variables [19]

sides into the OPC UA request formats then pass the requests toward the OPC UA server. The other way around is the same, where the response from the server is translated to the GraphQL format. Therefore, knowing how to access and exchange data with the GraphQL wrapper is sufficient for clients to communicate with the OPC UA server. Moreover, it is not required to modify the implementation of any existing OPC UA server when the wrapper is added to a new client system. As the wrapper is not combined with any specific server, it is also possible to aggregate several OPC UA servers within the same GraphQL API.

The GraphQL API supports various operations on nodes, including reading, writing, and adding new nodes with values, as listed in figure 47.

Data from OPC UA Server are typically fetched by the wrapper through batched queries. This means that clients can make a single query for the server, in the form of batching and combining several node attributes.

Since a session of retrieving data cannot be formed instantly, it often takes a longer

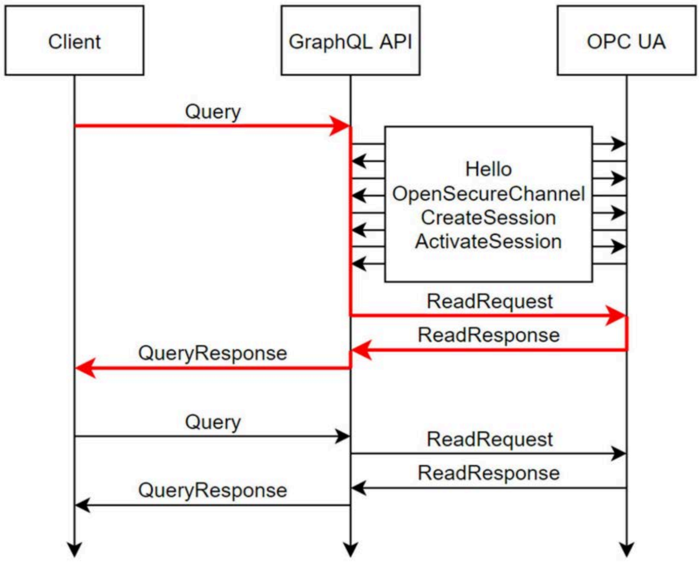


Figure 46: Communication Flow with GraphQL Wrapper [15]

Read	**Write**	**Add**
DisplayName Description Variable • DataValue • DataType • SourceTimestamp • Statuscode NodeId Child nodes	DataValue Description	Folder node Variable node

Figure 47: OPC UA GraphQL Wrapper Node Operations [15]

time of the first request in a session than of the queries following it up. Therefore, rather than start a new session, it is often a good practice to utilize an existing one, which could decline the networking traffic and improve the communication latency.

Besides, a web-based user interface with the GraphQL wrapper (see figure 48) has been developed together with the wrapper, to demonstrate how the wrapper could be utilized to ease the development process of an external application that communicates with the crane. The web app not only enables the control of crane movement through the web panel but also can display the crane status on the dashboard when corresponding OPC UA server nodes are selected.

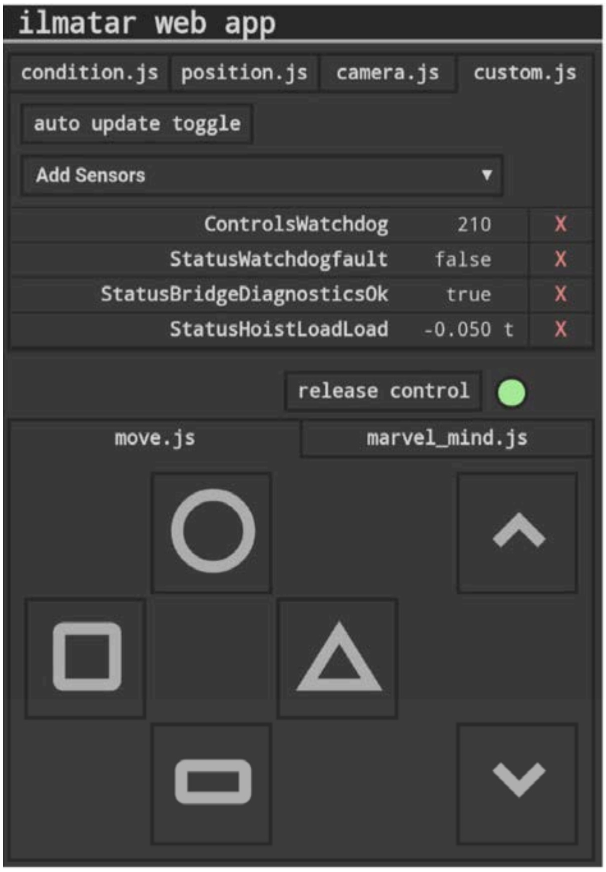


Figure 48: Interface of “Ilmatar” Web App with GraphQL [15]

4.1.4 Crane Connectivity

The connectivity solutions of the “Ilmatar” crane system have been evolving constantly. As the crane has movable subsystems during operation, no wired connection is available. Instead, the crane is equipped with several Wi-Fi routers forming wireless networks of various purposes, as well as an LTE modem for the mobile connection, which together enables various external systems to access various crane systems [?].

Figure 49 illustrates the connection, where external clients are able to assess the crane GraphQL API, the web app as well as the OPC UA server by connecting to the crane network via Wi-Fi. The connectivity solution is composed of two parts, i.e. servers onboard the crane, and the external network hub. The servers onboard the crane include the OPC UA server and GraphQL API running on a Raspberry Pi. They are both connected to a Wi-Fi router through Ethernet cables, which are located inside an electrical box onboard the crane so that a decent connection quality is ensured. The external network hub, on the other hand, is based in the same hall (AIIC) as the crane and is connected to the same crane Wi-Fi network as the server onboard the crane. The hub is equipped with a Wi-Fi extender, allowing the network to be accessible by supplementary systems or devices. For instance, the web application described in the previous section, which runs on a Raspberry Pi, is connected to the Wi-Fi extender through an Ethernet cable.

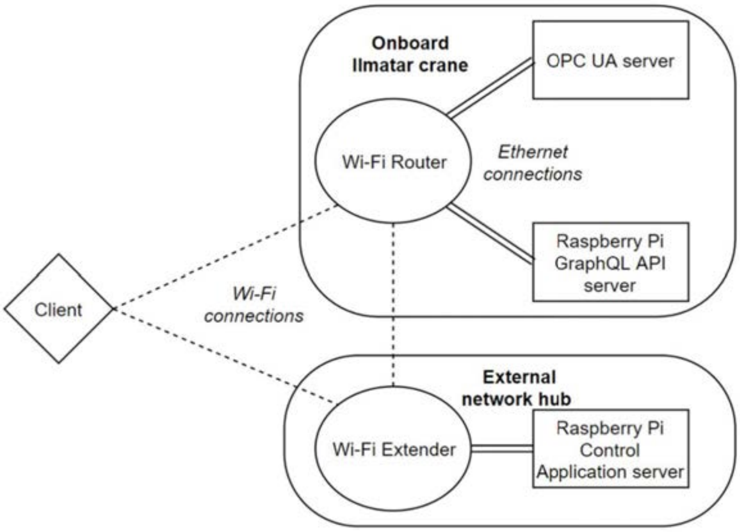


Figure 49: Crane Connectivity and Network [15]

4.2 Software and SDKs for MR Development

4.2.1 Unity 3D Engine and Mixed Reality Toolkit (MRTK)

Unity 3D Engine Overview

Unity is a cross-platform game engine developed by Unity technologies, which can be used to create three-dimensional, two-dimensional, VR/AR/MR games as well as many other experiences.

Unity is written in C++ for runtime and C# for Unity Scripting API, where the MonoBehavior is the base class for each Unity script to derive from. Several functions from MonoBehaviour that are critical in the project structure design are listed as follows [51].

- Awake() is called when the script instance is being loaded;
- Start() is called on the frame when a script is enabled just before any of the Update methods are called the first time;
- Update() is called every frame if the MonoBehaviour is enabled;
- FixedUpdate() is called every fixed frame-rate frame if the MonoBehaviour is enabled.

MRTK and its Input System

The Mixed Reality Toolkit (MRTK) provided by Microsoft empowers the in-Unity development of MR applications running in Microsoft HoloLens, where various HoloLens device capabilities are utilized, e.g. gaze, gesture and voice input systems. Figure 50 provides an overview of the MRTK Input systems.

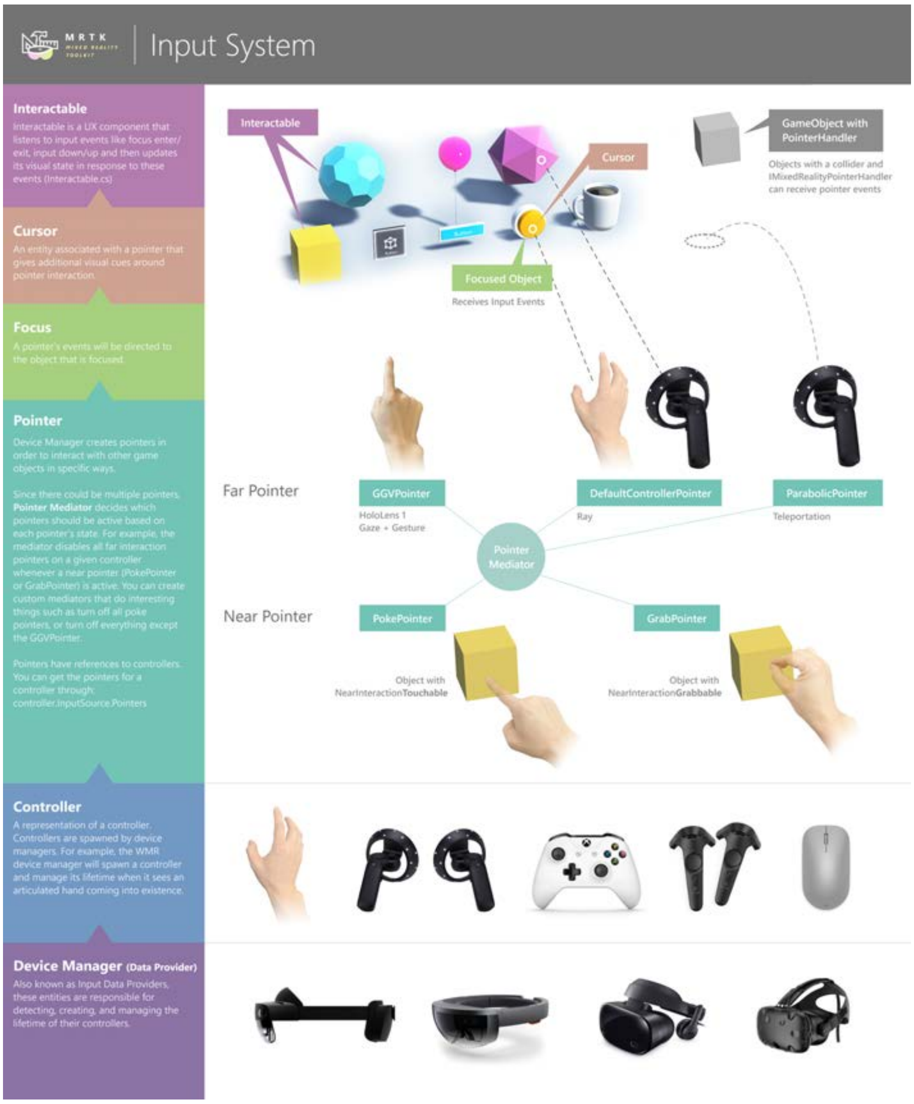


Figure 50: MRTK Input System Overview [26]

Inputs are produced by the Device Managers (Data Providers) such as the Microsoft HoloLens, with each provider associated with a certain input source such as the Windows Mixed Reality (WMR) controller. The in-use providers should be registered in the project profile within MRTK, so that the providers can automatically generate Controllers and produce Input Events once the corresponding input sources are detected. Pointers can be set up and attached to the Controllers, which query the MR scene, drive UI components and digital content via focus or Pointer Events [26]. Figure 51 illustrates the events flow.

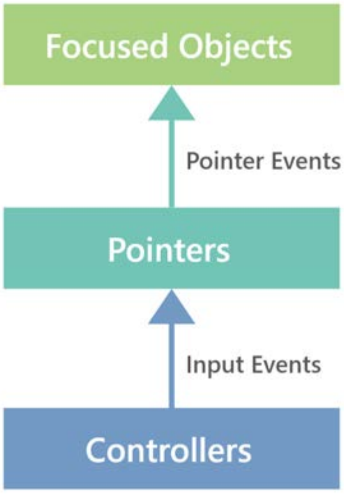


Figure 51: MRTK Input Event Flow [26]

4.2.2 Vuforia Augmented Reality SDK

Vuforia Overview

Vuforia is a software development kit (SDK) for AR applications, utilizing computer vision-related technologies to register or calibrate the digital content in the physical world, through real-time recognizing and tracking a variety of 2D and 3D target types, including “markless” image targets, 3D model target, and addressable fiducial marker named “VuMark”. In addition, Vuforia SDK features with the device localization with six-degree-of-freedom (DOF), localized Occlusion Detection with “virtual Buttons”, image target selection in runtime, as well as to programmatically create and reconfigure target sets in runtime,

Vuforia empowers HoloLens with a significant capability of connecting virtual objects and AR experiences with the physical objects in the real world. This capability, for instance, can be utilized to superimpose step-by-step instructions and procedural guidelines on corresponding machinery components or enhance the physical product, process, or system with digital details and gamification elements in factories or fields [57].

API in various programming languages are available in Vuforia SDK, such as JAVA, C++, Objective-C++ as well as .NET languages like C# via an extension to Unity. Vuforia SDK supports not only native app development for iOS and Android but also AR apps running in UWP with Unity game engine. Furthermore, existing Vuforia Engine mobile applications can be seamlessly configured in Unity into HoloLens apps running on the UWP, and the other way around.

Such a wide range of platform support for the optimal AR experience is enabled by Vuforia Fusion, which senses the capabilities of the underlying device and fuses them with Vuforia Engine features [54] (see figure 52).

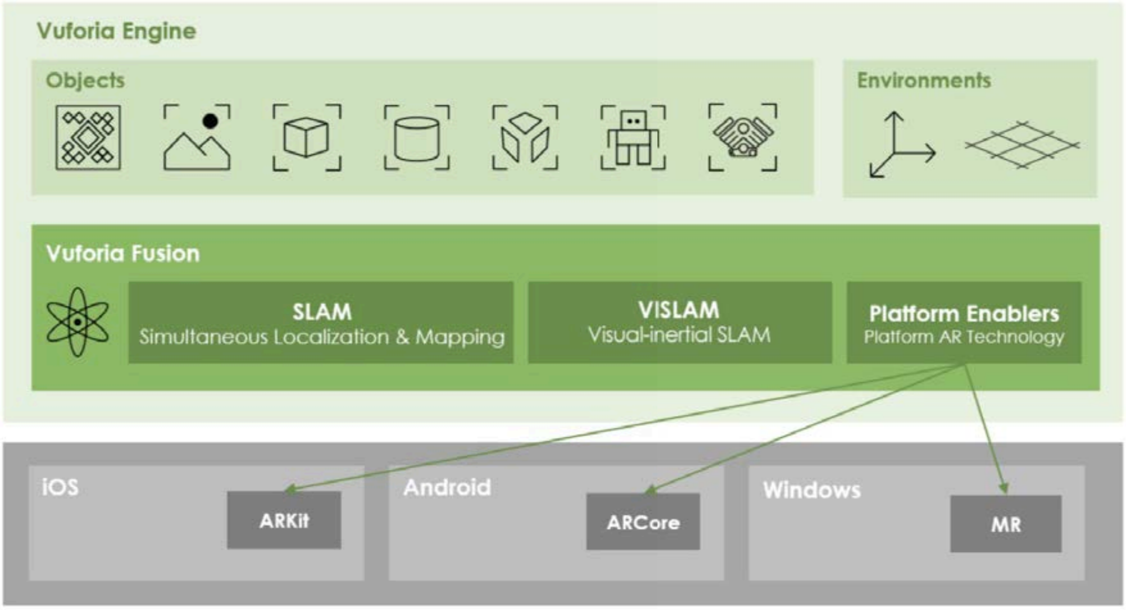


Figure 52: Vuforia Fusion [54]

Image Target

Image targets represent the images that can be detected and tracked by Vuforia Engine by comparing extracted natural features from the camera image with those from the resource database of the same target. Once the Image target is detected, Vuforia will track the image and augment the real-world scenes correspondingly by positioning and orienting the digital content with regard to the physical environment in the view's perspective [53].

Image targets can be adopted typically in gaming, visualizing products in its functioning environment, as well as recognizing and augmenting printed media and product packaging for marketing campaigns.

Device Tracking

Device tracking provides position and rotation updates of the device's location, which are computed in real-time from the sensor measurements and camera captures of the surroundings. Leveraging Vuforia Fusion, which detects and utilizes the native tracker of each platform, or the Vuforia sensor fusion technology, Vuforia Device Tracker is able to deliver robust and dynamic target tracking for various target types, platforms and devices [55].

The Positional Device Tracker, inheriting from the Device Tracker, utilizes visual details of the physical environment from the device camera view, as well as, if available, the built-in IMU sensors to detect the six-DOF pose of the device. This way, Vuforia could provide information on where the device is in relation to the physical world, to support a space-aware AR experience.

Extended Tracking

Extended tracking enables the availability of the target's pose information even when the target is out of the camera view or for some other reasons can not be directly traced anymore [55]. This, in practice, means that after device users turn their view away from the initial target, the holograms (augmented content) will still be rendered and sustain their positions with regard to the physical environment, as well as the consistency with the initial reference frame determined by the initially detected target (see figure 53). The quality of Device tracking performance can be improved by a feature-rich and highly-detailed environment.

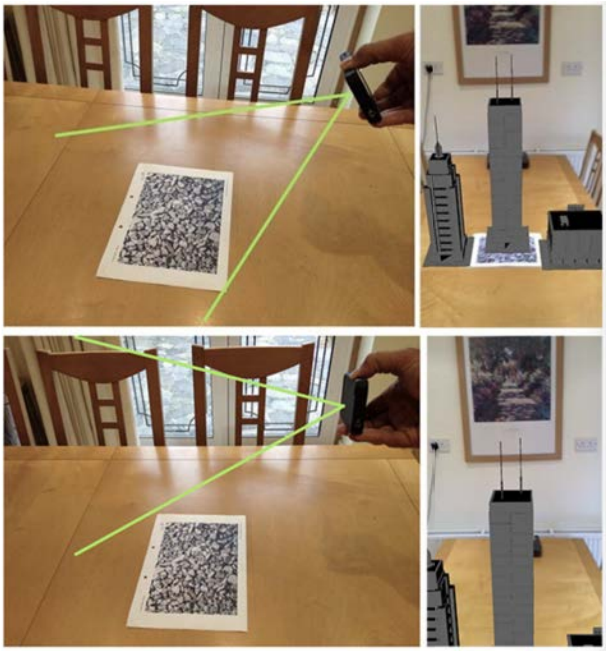


Figure 53: Extended Tacking Example [55]

With the HoloLens device, Vuforia Engine fuses the poses from camera tracking and HoloLens' spatial tracking automatically, thus providing stable target pose information without dependence on whether the device camera can view the target.

The process can be described in a high-level manner as follows [56].

- Target is recognized by Vuforia Target Tacker;
- Target tracking is initialized;
- The position and rotation of the target are analyzed to provide a robust estimation of the HoloLens pose;
- The target's pose is transformed into HoloLens spatial mapping coordinate space;
- Once the target is no longer located in the view, HoloLens takes over the tracking task, until the user looks again at the target, where Vuforia will

continue to track the images and objects or re-calibrate the space.

4.2.3 RestSharp Library

RestSharp is a comprehensive, open-source .NET client library for HTTP REST APIs, which is one of the most popular, with 32 million total downloads on NuGet and 10,000 average daily downloads [44].

RestSharp can be adopted to build robust applications thanks to its features that allow easily interfacing with public APIs and quickly accessing data with no need to directly handle complex raw HTTP requests. RESTful architecture offers an information-driven, resource-oriented approach for web application development, together with various configurable options like authentication, payload parsing, and URI generation. RestSharp supports both synchronous and asynchronous requests, making it possible to program on Windows and other platforms, as well as to create elegant applications that are easy to debug. Additionally, RestSharp is empowered with custom serialization and deserialization, automatic XML and JSON parsing, and feature supports like GET, PUT, POST, DELETE, OPTIONS, .etc.

4.2.4 Task Parallel Library (TPL) in .NET Framework

The Task Parallel Library (TPL) is a set of public types and APIs under .NET framework, which serves to simplify the process of adding parallelism and concurrency to applications [22]. The TPL scales the concurrency dynamically by using all the available processors in a most efficient way. Besides, the TPL is able to handle the work partitioning, cancellation support, state management, thread scheduling within a ThreadPool, and many other low-level tasks.

A common scenario to use the Parallel class is data parallelism, where the same operation is performed concurrently on different items of a source collection or an array [23]. The data parallelism scenario is supported by the TPL through System.Threading.Tasks.Parallel class, which offers method.based parallel implementation of “for” and “foreach” loops. The logic within the loops can be designed in a way that is the same as in sequential programming, as the TPL handles the low-level tasks automatically.

Behind the scenes, when a parallel loop starts to run, the TPL partitions the source data to enable the multiple threads to operate in parallel on different segments. Meanwhile, the Task Scheduler partitions the tasks with regard to system resources and workload, and later can also redistribute work among multiple threads and processors if the workload unbalance occurs.

Several overloads in both the parallel “for” and “foreach” methods enable the developers, for instance, to break loop execution, monitor the loop status of other threads, maintain thread-local state, finalize thread-local objects, as well as control the degree of concurrency. The Parallel.for method is used in this thesis for the concurrency within Unity script.

5 Prototypical Development

This chapter firstly depicts the overall architecture of the prototype, then highlights the modular design of the MR application, with a detailed explanation of each module's functionalities and workflow from users' perspectives, as well as the principle and logic behind the scene.

5.1 Prototype Architecture Overview

Figure 54 illustrates the prototype architecture, where the user can interact with the "Ilmatar" crane in a bi-directional manner, i.e. controlling and monitoring, through the MR application using the Microsoft HoloLens 1 device.

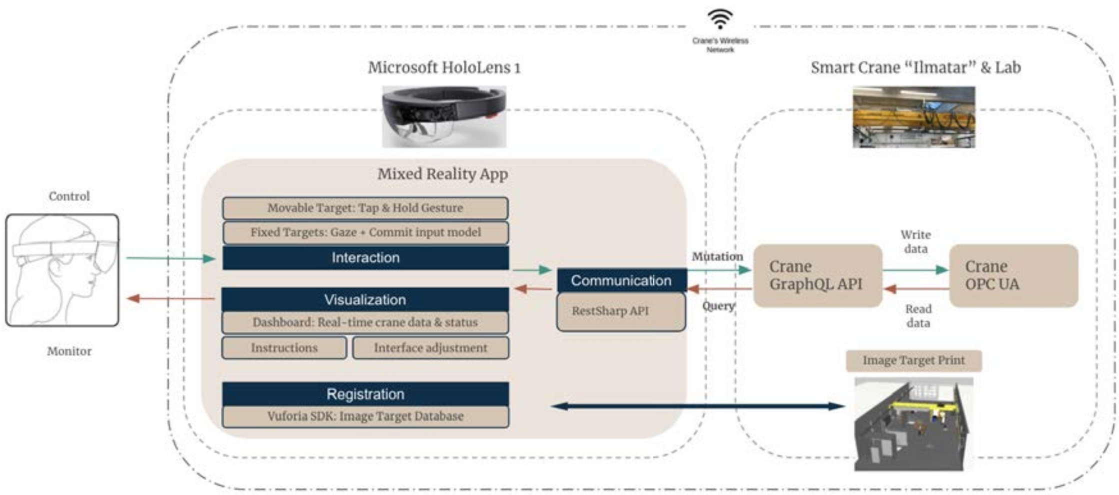


Figure 54: Prototype Architecture

Within the MR application, functionalities are divided into four modules, i.e. interaction, visualization, registration and communication.

On one hand, the user can control the crane through the interaction module with either of the two ways, i.e. fixed targeting control and movable targeting control. On the other hand, through the visualization module, the user can view the instructions on how to use the application, change the interface, and, most importantly, monitor the crane status. The registration module registers the virtual content with the physical lab environment, utilizing Vuforia SDK and an image target database within the MR application, together with the printed image target located at the real AIIC space. Through the communication module leveraging RestSharp API, the application can send HTTP requests in the form of either mutation or query to the crane GraphQL wrapper, in order to write or read data into the crane OPC UA. All the functionalities are developed heavily based on Unity 3D Engine with MRTK and its input system. Additionally, the control and monitor logic is divided into several threads and running in a parallel manner, which utilizes the TCL in the .NET framework.

The connectivity between the MR application and the crane “Ilmatar” is enabled by the crane wireless network, with which both components are connected.

5.2 MR Application Modular Design

As described in the section above, the MR application of the prototype is designed with consideration of modularization concepts, which enable the solution development of best practice and ease the adaptive process for new MR applications. Additionally, modular design also makes the maintenance or update procedures more flexible, such as the ad-hoc installation, hardware uninstall / removal, re-calibration and registration, .etc.

The following content illustrates detailed functionalities of each module in the MR application together with a necessary explanation of the logic design, principles, development or Unity project architecture behind the scene.

5.2.1 Interaction Module

The interaction module enables the control of the crane, through interacting with target virtual balls in the scene. There are two ways available for interaction, i.e. movable target control, and fixed target control, whose difference lies in whether the assigned crane’s target position is hooked to space or not. Figure 55 shows how the fixed and movable targets look like from the HoloLens view before any commit happens.

Figure 55: HoloLens View before Interaction

Functionality: Fixed Target Control

For fixed target control mode, there are four fixed targets, which are displayed as four yellow 3D sphere-shaped holograms of 20cm-diameter size, with the title "fixed target 1/2/3/5" on top of them, located at different locations of the AIIC space. HoloLens users can select any of them by using the gaze combined with an air-tap gesture. Once a target is select, it changes color from yellow to red as an indication of the successful commit. As introduced in section 4.1.2, if meanwhile the certain button in the crane manual controller is constantly pressed, and the access code and watchdog value are set as required, the crane should move to the selected target position.

Functionality: Movable Target Control

Movable target control works similarly as the fixed target control, with the only difference that instead of choosing from a fixed range of targets, users can drag and drop to freely define the location of a movable target in the space. The movable target is presented as a 20cm-diameter green 3D sphere. Once it is firstly selected using gaze, air tap and hold gesture, the sphere changes its color to red, and a spatial mapping shows in the scene, then the user can navigate the sphere in space while keeping the hold gesture. Once the user commits again with an air-tap gesture, the target is dropped and placed at a new location with its color turning into blue and the spatial mapping vanishing. With the same operation specified in section 4.1.2, the crane should then start movements towards the target position.

Figure 56 shows the HoloLens view while users are navigating the movable target, where the spatial mapping of AIIC space captured by the HoloLens camera is also displayed in the scene, in order to enhance users' spatial awareness and reduce the risk of placing the movable target at inaccessible locations.

Figure 56: HoloLens View during Movable Target Control

Logic of Target Control

Both of the interaction scenes are enabled by the MRTK and its input system within Unity 3D engine, which includes support for the gaze, air-tap and hold gesture inputs (see section 3.1.3).

The logic of target control behind the scene is that one script is constantly computing the value difference between the crane's current position and the selected target position of each subsystem (i.e hoist, trolley and bridge). According to the difference, the crane's trolley and bridge will then first move forward / backward until the difference is under a certain threshold. Out of the operational safety, only once the trolley and bridge stop movement, the hoist then starts to move up/down until the crane finally arrives at the target location in the three dimensions.

The current crane status used for computation comes from the communication with the GraphQL server (details see section 5.2.4), while the target positions are determined by a whole procedure of pre-calculation of the offset for transformation. To explain in more details, after the registration step (see section 5.2.3), with the holograms of fixed targets and movable targets appearing at their registered initial locations, manual control and calibration with eyes are conducted, in order to align the crane to each of the targets and note down the crane's actual positions of the hoist, trolley and bridge. These notes are then used to make comparison with the value of the holograms' position (i.e. transformation x, y, z in the Game Objects of four fixed targets) in Unity, and calculate the linear relations between the positions under the crane default system in the physical world and the ones in Unity scene. The scale and offset of this relation are determined by taking the average of the calculation on four target sets. The calculated linear relation is then utilized to transform the target positions in Unity and determine the actual target positions that are written into the crane system through communication module.

5.2.2 Visualization Module

The visualization module enables the users to directly monitor the crane real-time status through a dashboard, view the instructions on how to use the application, as well as adjust the interface (MR scene) in the way that they prefer through switching on/off the toggles of certain virtual objects.

The components in the visualization module are constructed utilizing UX building blocks from MRTK to make them visually appealing and operationally interactive.

Functionality: Dashboard

Figure 57 is the screen capture of the dashboard from the Unity 3D project in the play mode.

Figure 57: Dashboard in Unity Scene

The crane real-time status data displayed on the dashboard is composed of three parts of information, which are organized in a grouped manner with thoughtful design consideration on the layout.

The first one is the crane status, which includes its load value (kg), sway control feature (on/off), positions (m) and movements (for trolley and bridge stop/forward/backward, for hoist stop/up/down) of all the three subsystems (i.e. hoist, trolley and bridge). The data of this part is fetched from the GraphQL server (details see section 5.2.4).

The second part is the crane target, where the name (fixed target 1/2/3/5 or movable target) of the selected target and the position (m) of the target's three subsystems are shown. The data of this part is fetched directly by reading the corresponding parameters in the scripts attached to the Game Objects of fixed target and movable target (details see section 5.2.1).

The third part is the communication status, which shows if the behaviours/functions (i.e. watchdog, monitor and control) running in three different threads have started, as well as the updated value and status (ok/failed) of the watchdog. The data of this part is from the multi-threaded behaviour of the communication script (details see section 5.2.4).

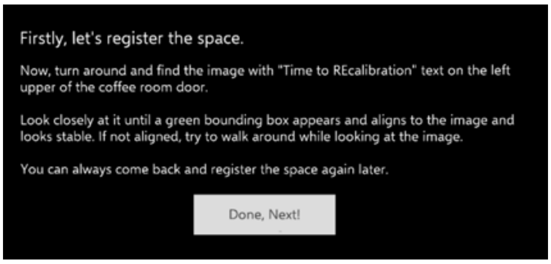
In the Unity project, the dashboard as a Game Object is placed under the "HoloLens Camera" hierarchy, meaning that it will always be displayed right in front of the user (HoloLens device camera) at a fixed distance so that the users can check the information easily wherever they are moving towards and look at.

Functionality: Instructions

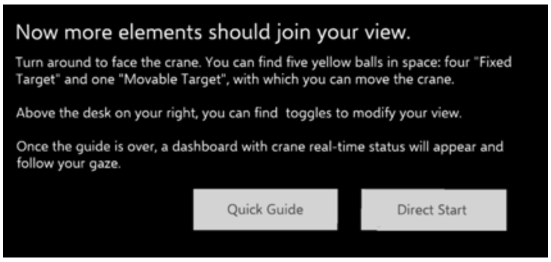
In order to improve the user experience (UX), make the interface friendly for new users without requiring a steep learning curve, the MR application is empowered with interactive instructions on how to use the application (see figure 58).

When the user starts the application, the welcome page (see figure 58a) comes first, followed by detailed information on how to register the space (see figure 58b). After the users confirm the completion of the registration step, the main hologram components of this application firstly appear in the scene, including the fixed and movable targets, dashboard, and scene-switch toggles. The next page (see figure 58c) then introduces what and where those new holograms are. From this page, users can choose to either check a "quick guide", which consists of three pages (see figure 58d, 58e, 58f) on what you can do with the application (i.e. monitor and control the crane, adjust the scene, as well revise the guide), or skip the guide and directly use the application. Figure 59 shows how the instruction looks like from the HoloLens view.

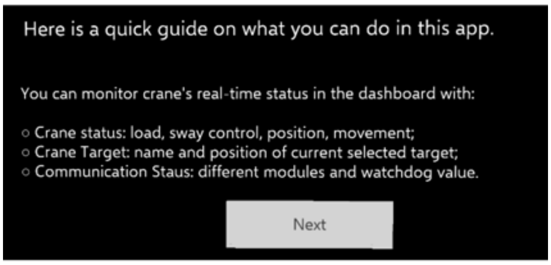
(a) Welcome Page



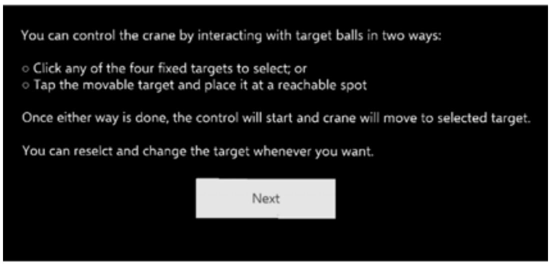
(b) Registration Instruction Page



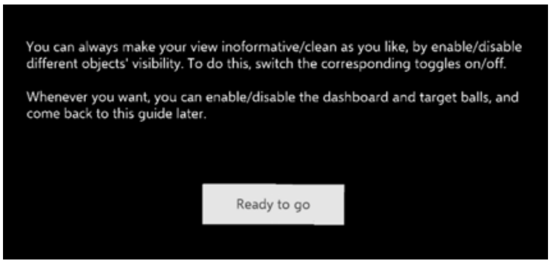
(c) Hologram Explanation Page



(d) Quick Guide Page 1



(e) Quick Guide Page 2



(f) Quick Guide Page 3

Figure 58: Instructions in Unity Scene

Figure 59: Instruction in HoloLens View

Functionality: Interface Adjustment

As described above, during the instructions, several holograms appear and stay in the scene, which provide users with components to interact with and information to view (see figure 55). However, too much information and unnecessary components for the operation of certain time-being could potentially affect user experience, and even disturb the crane operation.

For this reason, the functionality of the interface adjustment is introduced to this prototype. Figure 60 shows the scene-switch toggles, with which users can switch off the hologram components (i.e. dashboard, movable and fixed targets) in the default scene to clear up the view, and enable them again any time as users prefer. Additionally, the three pages of "quick guide" (see figure 58d, 58e, 58f) can be displayed again whenever users need a recall on how to use the application.

Figure 60: Interface Adjustment in Unity Scene

Development of Interactive UI

To construct the interactive UI, the Unity project utilizes the prefab UI from MRTK, such as the collision-based buttons and toggles, which are already configured to have audio-visual feedback for various types of inputs, including gaze and air-tap commit [29].

The events exposed in the prefab itself as well as the *Interactable* component are used to trigger additional actions. For instance, the pressable buttons used in the instructions leverage *Interactable*'s *OnClick* event to disable the presence of the current instruction page and enable the Game Objects of the next instruction page or holograms. The toggles used in the interface adjustment leverage *Interatable*'s *OnSelect* / *On Deselect* event to enable / disable the Game Object of corresponding holograms (see figure 61).

5.2.3 Registration Module

The registration module aims at placing the holograms at the pre-defined positions in the physical environment regardless of the user's location when the application is initialized. This way users do not need to always start the program from a certain location at the lab but benefit from improved mobility and flexibility.

Registration Procedure for Users

As described in section 5.2.2, the second page of instruction guides users to complete the registration procedure at the very beginning of the program.

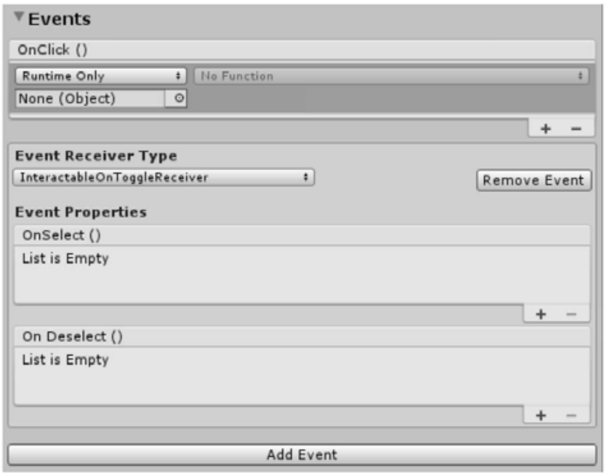


Figure 61: Toggle Event Receiver in Unity Scene

In details, users should turn around and find the printed image target with "Time to REcalibration?" text near the door at the AIIC space, and then look closely at the image target until a green bounding box appears and aligns to the image target stably. The holograms of the fixed and movable targets then appear at the pre-defined locations in the space. Figure 62 shows how the printed image target and its virtual bounding box look like from the HoloLens view.

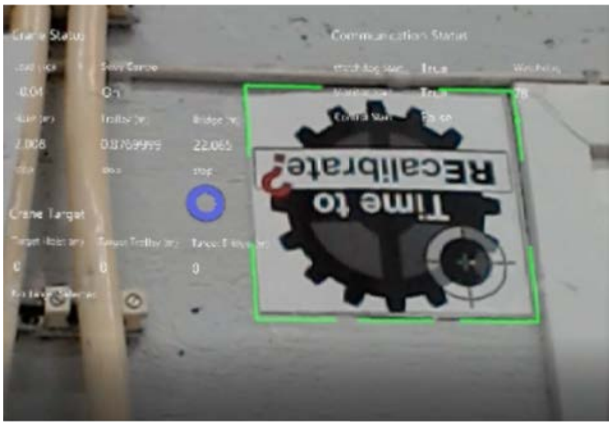


Figure 62: Image Target with Bounding Box in HoloLens View

Note that the registration step can be done any time later, in case the tracking is accidentally lost or the holograms are recognized as not appearing at the locations where they are supposed to be.

Development of Registration Module

The registration module is built with the Vuforia AR SDK in Unity, utilizing its features of image target, device tracking as well as extended tracking (see section 4.2.2). The details of the development are described as follows.

Firstly, an image target is selected and uploaded to the Vuforia developer portal to generate the image target database. The database is then imported to the Unity project and serves in the Game Object of Vuforia Image Target. The configuration of Vuforia in Unity is conducted to make sure the device tracking and extended tracking features are activated. On the other hand, the same image target is printed without changing its scale. Special attention is paid to making sure that the setting of database size in Unity is in correspondence with the actual size of the printed image.

Meanwhile, by scanning the space with HoloLens device, a spatial mapping mesh of the AIIC is retrieved from the Device Portal and imported into the Unity Project.

Next, the printed image is stuck firmly to the physical AIIC space, while its corresponding image target in the Unity project is placed with approximately the same pose (i.e. position and orientation) relative to the AIIC spatial mapping mesh. Figure 63 shows the holograms, image target and the spatial mapping mesh in the Unity scene. Comparing figure 63 with figure 62, it can be observed that both real and virtual image targets are located on the wall at the corner of the door frame. This step can be re-done by checking the alignment between the spatial mapping and the physical environment in the HoloLens view, and then adjusting the image target's relative pose in the Unity project.

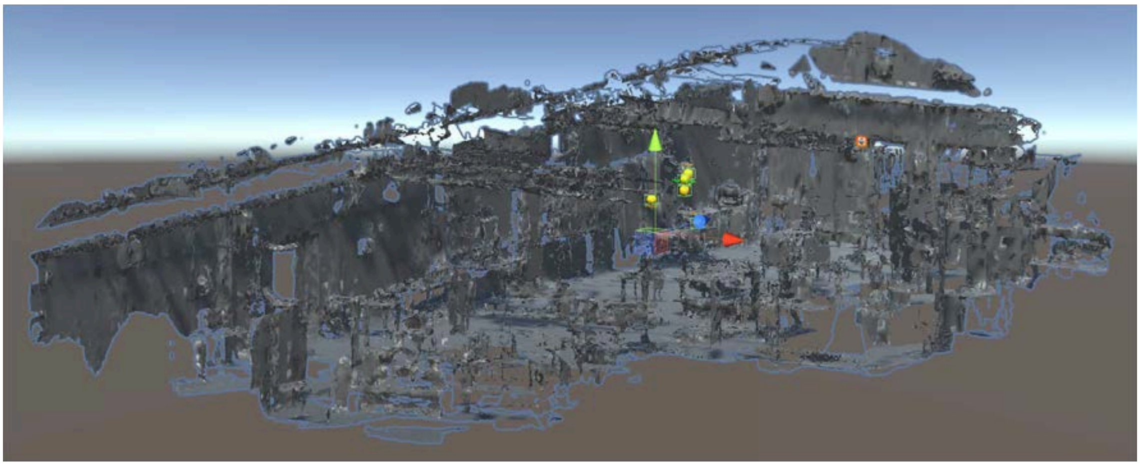


Figure 63: Spatial Mapping, Holograms and Image Target in Unity Scene

Note that spatial mapping only serves for development use. Although it together with the holograms of fixed and movable targets is organized as the child of the image target in the Unity hierarchy, the mesh just functions as a spatial reference that could help developers to quickly perceive where the other holograms are approximately located in the real world, in order to further adjust them easily. Therefore, once the scene building is complete, the spatial mapping is disabled and eventually does not in the deployed version of the application.

Besides, the hologram of a bounding box is placed around the image target in Unity project, in order to help users understand if the registration procedure is well completed in the initial steps.

Applied Spatial Relationship and Frame of Reference

Although the general principles of tracking and registration have already been explained in section 2.4, figure 64 illustrates the spatial relationship in the registration module.

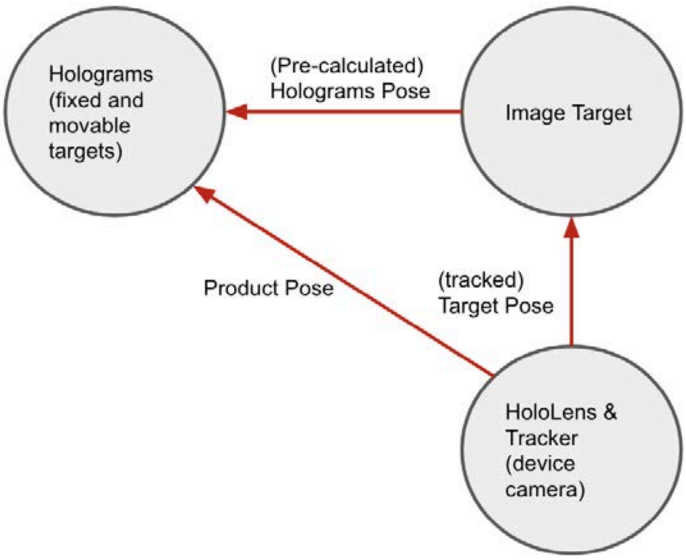


Figure 64: Full SRG in Registration Module

As shown in this SRG, the poses of the fixed and movable targets holograms are pre-calculated with respect to the image target (see section 5.2.1), while the HoloLens with the device camera can function as a tracker that is able to access the pose of the image target constantly with Vuforia extended tracking. By concatenating the holograms pose relative to the image target, and the image target pose relative to the HoloLens, the holograms pose with respect to the HoloLens can be derived.

Three major frames of reference (FOR) and their coordinate systems (CS) in the Vuforia Engine are applied in this registration module (see figure 65).

- World FOR: it is defined as the start position of the device, which is the reference origin for the AR interaction, with its CS right-handed, Y-up and gravity aligned;
- Camera/Device FOR: with its origin on the physical device (or more precisely, the camera position on the device), the camera/device pose are reported in the World CS, with its CS right-handed and Y-up;
- Target FOR: used for tracking physical objects or environments locations, the

target poses are reported in the World CS, with its CS right-handed, but up convention varying with dependence on the types of the trackable targets. For instance, the CS of the image target is Z-up.

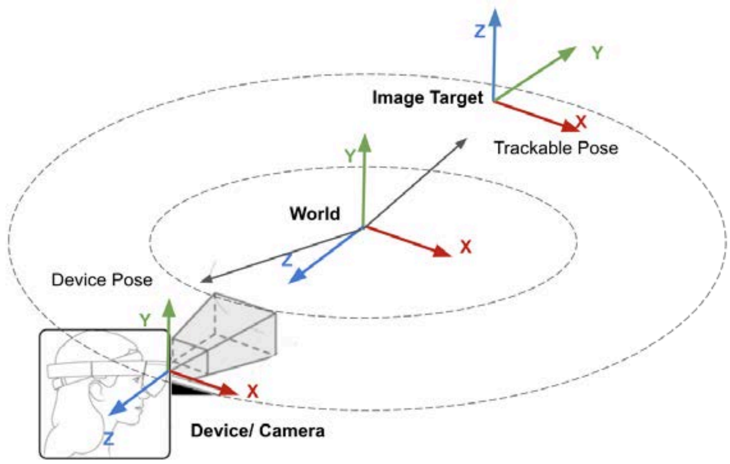


Figure 65: FOR of World, Device/Camera and Targets

5.2.4 Communication Module

The communication module functions as a bridge between the crane and all the other modules in the MR application prototype. It leverages the RestSharp API introduced in section 4.2.3 to send HTTP request (either query to read data, or mutation to write data) to the crane GraphQL wrapper (see section 4.1.3). This way the application can access the crane data specified in section 4.1.2, which can be further displayed in the dashboard of the visualization module, or modified in the target control of the interaction module.

Multi-Threaded vs. Single-Threaded Behaviour

As discussed in section 2.3.3, parallel computing technique could be potentially beneficial in terms of maximum utilization of multiple CPUs of the HoloLens device. Therefore, multi-threaded behaviours are implemented utilizing the TPL in .NET Framework introduced in section 4.2.4. To be specific, the parallel.for function is used to create loop logic for going through each thread. This way, the functions of monitoring, controlling as well as the control setup are running on three separate threads. The logic within each thread is illustrated as follows.

- **Watchdog Thread**: The watchdog thread enables the control module by setting the initial access code and updating the watchdog value constantly, which are both included in one mutation request body;
- **Monitor Thread**: The monitor thread enables the continuous access to the selected crane real-time data, including all the crane status data displayed on

the dashboard indicated in section 5.2.2, which are all included in one query request body;

- **Control Thread**: The control thread enables the interaction module by taking the difference between the current location and the target location then computing the movement speed and direction of each crane subsystem, which are all included in one mutation request body.

As a comparison, another single-threaded version of the communication script is carried out, with the functionalities of watchdog, monitor and control running sequentially in one thread.

Data Flow with Communication Module

Figure 66 illustrates the data flow between the communication module and the other modules as well as the GraphQL server.

The interaction module can give the signal to call the watchdog function, and determine the target data as the input flowing to the control function, while the visualization module can receive the crane status data flowing from the output of the monitor function, and use it on the dashboard. On the other hand, both the watchdog function and the control function send mutation requests while the monitor function sends queries to the crane GraphQL server, which together enable the application to exchange data with the crane in a bi-directional manner.

Determining Request Frequency, Size and Timeout

For either multi- or single-threaded behaviour, the three main functionalities are placed within the Unity MonoBehaviour FixedUpdate(), which is called every fixed frame-rate frame if the MonoBehaviour is enabled (see section 4.2.1). The frame rate can be adjusted in the Unity general setting to change the frequencies of HTTP requests, which is critical for communication performance.

On the other hand, the size of the request, in terms of how much data included in one request body, as well as the timeout, which indicates the time to wait before the request times out, are also adjustable and important parameters for communication.

Therefore, various combinations of the request frequency, size and timeout are carried out as different experiment settings in the prototyping process. Analytical consideration and qualitative evaluation are applied in iterative development.

For instance, too low request frequency can cause insufficient watchdog value update, thus stopping the control module from functioning. The frequency can also affect the real-time performance of the visualization and control module, which means that with too low frequency, the crane data shown in the dashboard can be outdated, and it can take a too long time for the crane to respond to control instructions, which will results in the delayed movement at the first place, or moving back and forth while approaching the target position. On the other hand, too high request frequency can

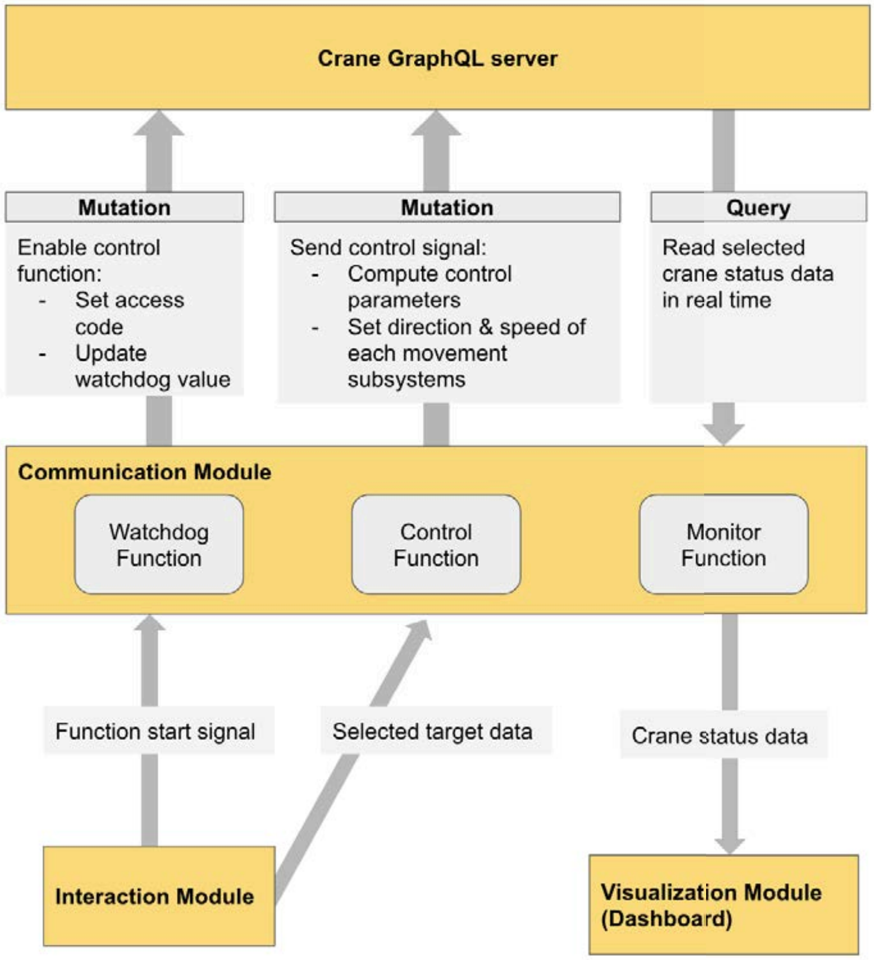


Figure 66: Data Flow with Communication Module

overload the device capability and even freeze the running application, with flicking holograms occurring.

There is also a trade-off between the size and frequency of the request. Generally speaking, using many small-sized requests is often better than using a few big-sized requests. However, in an unstable network environment, the latter can also be a good robust practice.

The timeout value is also modified other than the default timeout of 100 seconds, since in unstable network, too long waiting time delays the next request, thus affecting the real-time performance of the communication module.

Overall, the selection of request frequency, size and timeout is case-specific and no universal rules available to apply. For this application and under the current network environment of AIIC, the frame rate is set as 100ms and timeout as 1s. Regarding the request size, each request contains all the data needed for that functionality.

6 Evaluation

This chapter describes the evaluation process and evaluation results of this application, which focuses on the accuracy of fixed and movable target control methods.

6.1 Evaluation Procedure

This section introduces the approach applied in the evaluation procedure, covering how to collect data, calculate error based on data, and visualize the evaluation result.

6.1.1 Collecting Evaluation Data

To carry out the evaluation, a reversed approach in contrast to the procedure of determining the translation described in section 5.2.1 is conducted.

In detail, once space is registered, the user selects a fixed target in the HoloLens scene within the application, then manually navigates the crane with a remote controller to the selected target, where the crane hook is aligned with the hologram of the target. Then the target positions P_{target} (predicted value) and the crane status positions P_{status} (observed value) of the three subsystems are read from the hologram dashboard and recorded.

6.1.2 Calculating Control Accuracy

Such a procedure is performed for each of the four fixed targets and the movable target with an arbitrary moving. Then space is registered again before repeating the next test sets, in order to take into account the variation caused by registration. In total, fixed target control is evaluated with 3 experiments, thus 12 available test sets. On the other hand, 12 experiments are set for movable target control.

The data collected is used to calculate the Root Mean Square Error (RMSE) with the following formula, where the value of n is 12 for fixed target control, and 10 for movable target control.

$$RMSE = \sqrt{\frac{\Sigma_{i=1}^{n}(P_{target} - P_{status})^2}{n}}$$

6.1.3 Visualizing Evaluation Data

To illustrate the distribution (clustering) pattern of the collected data, the principal method of error ellipse, i.e. confidence ellipse, is utilized in the visualization. As all the three dimensions of the hoist, trolley and bridge need to be taken into consideration, a higher dimensional version of the error ellipse, called "minimum volume covering ellipsoid" (MVEE), is drawn for each of the control methods. In other words, two MVEEs enclose N points in a D-dimensional space, where the N is 12 for fixed target control, 10 for movable target control, and D is 3 for both methods.

The detailed procedure of generating the MVEEs can be divided into two steps.

Firstly, the center, radiis and rotation of the MVEE are determined utilizing the Khanchiyan algorithm. The meanings of each parameters are interpreted as follows:

- MVEE center represents the average error, taking all the data points into the calculation;
- MVEE radiis represent the length of the equatorial axes perpendicular to each other, indicating the variance;
- MVEE rotation provides the orientation of the ellipsoid.

Then, the resulting center, radiis and rotation are used to plot the ellipsoids utilizing python matplotlib library, where the final visualization is the generated ellipsoid mesh, with all the data points maintained and the three radiis highlighted.

6.2 Evaluation Result

This section presents the collected data, the RMSE calculation as well as the error ellipsoid plots based on the data.

6.2.1 Collected Data and RMSE Calculation

Figure 67 and figure 68 present the data collected from fixed and movable target control evaluation, as well as the calculated RMSE with respect to each movement subsystem, where "B" stands for the bridge, "H" for the hoist and "T" for the trolley. The unit is meter, the same as how they are like from the dashboard hologram.

Fixed Target Control	Test 1				Test 2				Test 3					
Target position	1	2	4	5	1	2	4	5	1	2	4	5		
B	20,684	20,04	20,523	20,04	20,684	20,04	20,523	20,04	20,684	20,04	20,523	20,04		
H	2,021	2,021	1,055	1,538	2,021	2,021	1,055	1,538	2,021	2,021	1,055	1,538		
T	1,561	1,883	1,883	1,078	1,561	1,883	1,883	1,078	1,561	1,883	1,883	1,078		
Status position	1	2	4	5	1	2	4	5	1	2	4	5		
B	20,703	20,06	20,533	20,066	20,765	19,998	20,606	20,099	20,778	20,196	20,51	20,025		
H	2,024	2,024	1,073	1,529	2,015	1,992	1,074	1,53	2,005	1,99	1,049	1,527		
T	1,583	1,914	1,906	1,112	1,596	1,91	1,926	1,12	1,602	1,959	1,951	1,142		
Difference	1	2	4	5	1	2	4	5	1	2	4	5		RMSE
B	-0,019	-0,02	-0,01	-0,026	-0,081	0,042	-0,083	-0,059	-0,094	-0,156	0,013	0,015	Bridge	0,05508251
H	-0,003	-0,003	-0,018	0,009	0,006	0,029	-0,019	0,008	0,016	0,031	0,006	0,011	Hoist	0,01607016
T	-0,022	-0,031	-0,023	-0,034	-0,035	-0,027	-0,043	-0,042	-0,041	-0,076	-0,068	-0,064	Trolly	0,04552838

Figure 67: Fixed Target Control Evaluation Data

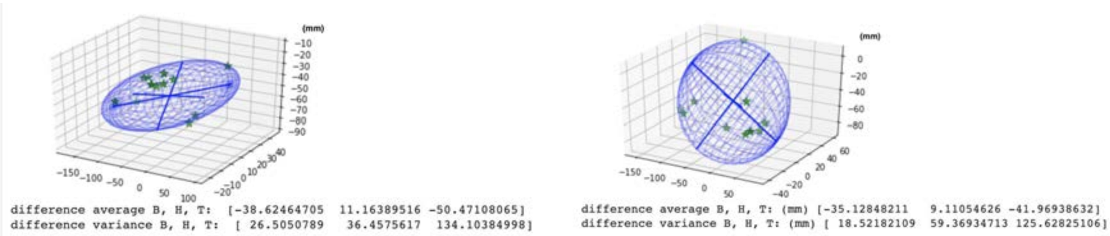
Movale Target Control												
Target position	Test 1	Test 2	Test 3	Test 4	Test 5	Test 6	Test 7	Test 8	Test 9	Test 10		
B	21,367	20,771	20,85	21,177	21,927	21,748	21,181	21,535	22,176	22,383		
H	0,854	1,171	1,316	1,628	1,523	1,765	1,317	1,413	1,32	2,312		
T	1,065	1,889	1,424	1,855	1,448	2,259	2,683	2,701	2,19	1,472		
Status position	Test 1	Test 2	Test 3	Test 4	Test 5	Test 6	Test 7	Test 8	Test 9	Test 10		
B	21,394	20,871	20,943	21,308	21,924	21,72	21,15	21,513	22,131	22,461		
H	0,86	1,187	1,322	1,628	1,522	1,755	1,317	1,427	1,344	2,256		
T	1,131	1,942	1,466	1,913	1,483	2,321	2,749	2,765	2,243	1,467		
Difference	Test 1	Test 2	Test 3	Test 4	Test 5	Test 6	Test 7	Test 8	Test 9	Test 10		RMSE
B	-0,027	-0,1	-0,093	-0,131	0,003	0,028	0,031	0,022	0,045	-0,078	Bridge	0,06847335
H	-0,006	-0,016	-0,006	0	0,001	0,01	0	-0,014	-0,024	0,056	Hoist	0,02082547
T	-0,066	-0,053	-0,042	-0,058	-0,035	-0,062	-0,066	-0,064	-0,053	0,005	Trolly	0,05352383

Figure 68: Movable Target Control Evaluation Data

It can be observed that the RMSE values for both methods are quite similar to each, within the range between 1cm and 7cm, where, at the same time, a similar pattern among all the subsystems, i.e. $hoist < trolley < bridge$, happens in both methods.

6.2.2 Error Ellipsoid Visualization

Figure 69 illustrates the generated error ellipsoid plots of fixed and movable target control separately in different graphs, where the error average (ellipsoid center) and error variance (ellipsoid radii) of each method are also computed and shown below the corresponding plots.



(a) Fixed Target Control (b) Movable Target Control

Figure 69: Error Ellipsoid Plots (separate)

To compare the error ellipsoids of two methods, another joint plot is generated (see figure 70) with blue mesh and stars for movable and the red ones for fixed target control. It can be observed that the ellipsoid of the movable target control is generally larger than the one of the fixed target control method, which is indicated by the length of the radiis. Meanwhile, the centers of the two error ellipsoids are close to each other, suggesting a similar distribution pattern and causes of the error behind the scene.

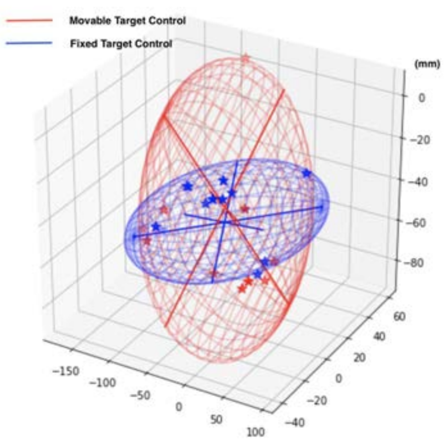


Figure 70: Error Ellipsoid plot (joint)

7 Conclusion and Outlook

This chapter concludes the thesis by reviewing and answering the research questions, where the challenges and solutions in prototyping and evaluation phases are addressed in a summative manner, leading to a high-level discussion. An outlook concerning the future work comes at the end of the thesis.

7.1 Answers to Research Questions

This section intends to answer the two research questions raised in section 1.3, which are shown again as follows:

1. How to develop a prototype of an MR interface for Digital Twin based crane operation? <discussed in chapter 5>
2. How to evaluate the performance of the MR interface in terms of the interaction accuracy with different control methods? <discussed in chapter 6>

Although the content of chapter 5 and chapter 6 can already serve as basic materials for answering the two research questions, respectively, this section provides answers from an alternative perspective in terms of the challenges we are faced with during the development process, as well as the corresponding solutions, which form a higher-level discussion accompanied with highlights of the work.

7.1.1 Q1: Prototyping - Challenges and Solutions

Challenges in Prototyping

The challenges of developing the prototype firstly lie in the fact that it is required to take into account the practices in the emerging fields of both industrial MR and digital twin related technologies, which are both under constant development and composed of various changeable elements. This brings complexity in terms that various components need to be taken into design consideration. At the same time, those components are highly dependent on each other, which requires a trade-off combination of features and functionalities that can be carried out in one POC prototype.

Furthermore, as the concrete case study of the smart crane “Ilmatar”, together with the corresponding research environment AIIC, is determined at the beginning of the work, the hardware in use is therefore restricted to the currently available resources, i.e. the capabilities of the crane platform and the HoloLens 1 device. For example, unlike the 2nd generation, the HoloLens 1 does not support complex gesture recognition and manipulation input, which needs to be taken into consideration while designing the interaction within the MR application.

The hardware setup further affects the associated software capabilities, such as the software system and connectivity solutions in the crane. For instance, the latency of the application highly depends on the network and the GraphQL server performance of the crane.

Meanwhile, the development tools are restricted to the common practice and most recent releases available in the market. This leads to the immature features of those tools and, particularly, the compatibility issues both in horizontal and vertical directions when different tools and packages need to be utilized and function together seamlessly in the prototype.

Another challenge in developing the prototype lies in utilizing the rich data sources from the crane Digital Twin platform and the corresponding potential to manipulate the data. This requires thoughts on which data to select, how data exchange could happen, and how to enable the MR application as an interface of data exchange between the user and crane.

The mobility of users is also critical. However, it is often affected by the fact that users have to start an MR application at a certain initial location in the physical space to ensure holograms are placed properly. To improve this, alternative solutions need to be investigated to replace the common practice of utilizing the world anchor or only the spatial relationship between holograms and HoloLens device (i.e the user wearing the device).

Last but not least, providing a user-friendly interface, especially reducing the difficulties for the first-time users of the application is also a challenge worth considering in the POC phase. A good UX is fundamental for application design and validation, and decisive for industrial MR to be put into actual adoption. Particularly, in the case study of “Ilmatar” crane, the UX should help the MR application to achieve the experience goals of effectiveness and empowerment for crane operators.

Solutions for Prototyping

To tackle these challenges in prototypical development, initial planning of modular design (see section 5.1) is carried out, together with an iterative trial-and-error development method applied behind the scene.

Along the prototyping process, the technical feasibility is discovered and validated constantly, while the capabilities of the software and hardware in use are explored and exploited, to push the potential and limitations of the prototype performance and functional variety. For instance, since the HoloLens 1 device does not support the "drag and drop" gesture like in HoloLens 2, an alternative committing method of "tap and hold" is discovered and utilized for interaction with the holograms of the movable target (see section 5.2.1).

Different combinations of development tools, packages and frameworks are also put into experiments, in order to resolve the compatibility issue and figure out an optimal solution of integrating various components and functionalities into one coherent interface.

Due to the dependencies on other components, some additional work is performed, such as fixing, debugging and redeploying the existing GraphQL server and troubleshooting the networking issue of the crane.

Meanwhile, the data exchange is well handled. On one hand, the data flow is designed between the communication module and the other modules within the application, as well as between the communication module and the GraphQL server (see section 5.2.4). On the other hand, the bi-directional information exchange is formed between the user and interface, where users can monitor the crane status and interact with the crane through the application.

Additionally, in order to improve the users' mobility, a registration module is integrated into the application, where the spatial transformations among holograms, space and HoloLens are configured, leveraging (extended) tracking on the image target. This way, the holograms are placed with pre-defined poses relative to the physical space regardless of the user's location when the application is initialized (see section 5.2.3).

To guarantee a good UX, the prototype leverages various UI interactive components in building hologram scenes, as well as takes many design considerations into its development details, which can be summarized as follows.

- In the interaction module, the holograms of the fixed and movable targets are presented with color changes when they are selected, which provide users a clear visual response while interacting. Additionally, while navigating the movable targets, the spatial mapping of the operational space is displayed in the scene, which serves as a reference for users to place the target with spatial awareness and thus empowering a safe operation(see section 5.2.1).
- In the visualization module,

  - The layout of the dashboard is well designed with crane status information organized in a grouped manner (see section 5.2.2 Dashboard);

  - The instructions, as the initial visual content of the application, guide new users through the application flow (see section 5.2.2 Instructions);

  - The interface adjustment function provides users with flexibility in adapting the virtual content according to their preference and real-time operational needs, thus forming a customized experience while using the application (see section 5.2.2 Interface Adjustment).
- In the registration module, a virtual bounding box would also be displayed in the HoloLens view, whose alignment to the actual image target shows users completion of the registration step (see section 5.2.3).

7.1.2 Q2: Evaluation - Challenges and Solutions

Challenges in Evaluation

The initial challenge of the evaluation lies in how to determine the values of the predicted/target position and the observed position for computing the control error/accuracy. It is not straightforward to figure out which data could serve this purpose in the first place.

Let's say we interpret the target/predicted position as the intention from a HoloLens users' viewpoints. In other words, users move the movable target to a certain point (target/predicted position) that they choose in the 3D physical space, then the observed position would be where the crane hook is located after the whole movements under the subsequent target control. However, if this is the case, the error needs to be measured manually with scale, as neither of the position values can be read directly from HoloLens. This brings with new challenges since the reference for measurement needs to be further determined and the manual measuring process often causes additional errors.

Subsequently comes the challenge of how to visualize the data, especially to highlight the pattern behind it and to compare the results of the fixed and movable target control methods. As either target/predicted and observed position values is composed of three dimensions in the hoist, bridge and trolley directions, the pure data is therefore featured with relatively complicated patterns for interpretation, not to mention comparing the patterns of the two methods and mining some insights out of it.

Solutions for Evaluation

Regarding the first challenge, a reversed logic for error measurement is applied, where the target/predicted value is the position of the selected target, and the observed value is the crane's real-time position status (see section 6.1.1).

Therefore, instead of determining the error by manual measuring the distance in physical space, both target/predicted and observed values can be directly accessed from the dashboard in the HoloLens view and then used to compute the error/accuracy with the unified RMSE formula (see section 6.1.2). This leads to a straightforward, clean and handy procedure of data collection and error calculation, which significantly empowers the whole evaluation workflow.

In order to guarantee a sound evaluation, the procedure of data collection is repeated 10 times for movable target control and 12 times for fixed target control, while RMSE calculation based on the collected data is conducted for each error dimension (i.e. hoist, bridge, trolley) and each control methods, thus 6 times.

The resulting sample data is well organized in the spreadsheet, with the RMSE values of three movement directions presented for each method, which enables various possibilities to explore the data either vertically or horizontally (see section 6.2.1).

Furthermore, an informative visualization is generated to illustrate the collected data and the clustering pattern beneath it, leveraging one of the error ellipsoid methods, MVEE (see section 6.1.3).

In the visualization, the average and variance of error are well presented by the center and raddis of the ellipsoid mesh, respectively. The ellipsoid for each control method, together with the sample data points, is presented separately in a space with three dimensions of the hoist, bridge and trolley. Additionally, a third visualization with the two plots jointly presented is also available, in order to compare the fixed and movable target control methods intuitively (see section 6.2.2).

7.2 Limitations and Future Work

As stated in the previous section, the work manages to tackle various challenges in prototyping and evaluating an MR application for the digital twin based crane, which sufficiently satisfies the goals of the POC phase. Yet within the limited time frame, many possible experiments, adaptions and tests have been left for the future. This section therefore addresses the limitations of the work and proposes some interesting research topics that worth investigating further as a continuation of this work in the future.

Robust Registration and Tracking

The control errors mainly come from the gap between the pre-calculated positions of the holograms (i.e. fixed and movable targets) and the holograms' actual locations in the physical space after the registration step. It is because the pre-calculation is based on the ideal conditions with flawless registration and tracking performance. However, the reality is that stable and accurate tracking information of the image target's pose cannot be guaranteed, thus resulting in spatial shifts of the holograms, which uses the image target's pose as a spatial transformation reference.

Therefore, in order to improve the control accuracy in terms of reducing the average error and variance, robust registration and tracking approaches could be applied in the work.

For instance, instead of only one image target, the work could adopt two and more registration markers with different poses and utilize multiple tracking information to generate a robust spatial transformation of the holograms poses. This way, the tracking error occurring specifically in a certain dimension of space could be potentially compensated by a more accurate tracking data of another marker. Furthermore, in case the device loses (extended) tracking of a certain marker occasionally, the tracking data from other markers could still serve the purpose.

Other possible techniques for registration and tracking could also be considered, such as utilizing the 3D model target of the crane hook, which is functionally supported by Vuforia SDK as well.

Latency Evaluation

Within the limited time frame, the POC prototype is evaluated from the control accuracy angle, which, however, only covers the spatial behaviour of the application. In order to integrate the temporal performance into the loop and form a complete quantitative evaluation, latency could be evaluated in future work.

Note that latency in this context can be interpreted with various meanings. For instance, it can represent the duration between the action of interaction (e.g. selecting the target hologram) and the execution of the crane movement. Alternatively, it can also be measured by the time difference between the execution of the HTTP request from the MR application, and the response (i.e. corresponding value change) in GraphQL server or the Crane OPC UA nodes. Different interpretations of latency would require different setups of the evaluation environment and different methods to collect and process corresponding data in need.

Device Upgrade

Furthermore, as HoloLens 1 device used in the work only supports limited gesture input types and some of its features are slightly outdated, future experiments could be conducted on more advanced and recently released devices, such as HoloLens 2. Along with the upgrade of the hardware, the project would also be adapted and enhanced with improved capabilities, which could potentially lead to more intuitive interaction and pleasant experience for application users.

Usability Testing

Due to the COVID-19 epidemic, the work does not cover any usability testing, neither at the initialization nor evaluation phase, but rather determine the UI design barely based on early empirical research as well as the common practice and criteria of HoloLens application usability.

However, usability testing is fundamental in HMI and UX design. Future work should integrate it into the whole MR development loop, from conception to iterative prototyping until the final evaluation.

Digital Twin Data Utilization

The work only utilizes the real-time data from the crane OPC UA through communication with the GraphQL server. However, along with the constant development of the crane's digital twin platform, data of various formats and from different resources is becoming easily accessible, which includes historical data of the crane usage, as well as the 3D simulation model of the crane and the operation environment.

In the future work, a bigger variety of data from the digital twin platform could be selected and integrated into the MR application, either by directly displaying it in the dashboard, or interpreting and visualizing it in the form of interactive holograms.

Related research topics would also be involving Artificial Intelligence (AI) in digital twin data processing, such as leveraging machine learning techniques on time-sequential crane operational data, or computer vision technology on the crane/HoloLens camera captures. Those advanced AI-enhanced features could introduce more innovative elements into the MR application, and bring the data utilization to another level.

Functionality Enhancement

At the conception phase of the work, different use cases for the smart crane HMI were investigated in several brainstorming rounds, leading to a comprehensive UI specification list (see figure 71). However, only several core representative features out of them were carried out in the POC prototype within the limited time frame of the thesis.

Category	Feature	Tool	Resource	UX
Visualization	Crane Status Dashboard	MRTK	Real time/ historical crane Data	Effectiveness
	Warehouse Holomap		Lab CAD Model	Effectiveness
	Mesh Space Viewfinder		Lab CAD Model	Effectiveness
	Safety Zone Indicator		Lab CAD Model	Safety
	Directional Guidance (Wayfinding)		Lab CAD Model	Effectiveness
Control	Gaze, Gesture and Voice Control	MRTK	Real time crane data	Effectiveness, Empowerment
	User Attention Directors	Spatial Sound	Real time crane Data	Safety
AI-enhancement	Task List Suggestion	Machine Learning	historical Crane Data	Effectiveness
	Gravity Point Indicator	Computer Vision	HoloLens Camera	Effectiveness
	Object Detection	Computer Vision	HoloLens Camera	safety
	Daily Report		historical Crane Data	Completion, Empowerment
Communication	Remote Expert / MR Experience Sharing	Vuforia Chalk	/	Fellowship
	Collaboration Tools	Vuforia Chalk, multiple Digital Twin	/	Fellowship

Figure 71: UI Specifications

With this POC prototype and the development experience functioning as the basis, more features could be taken into experimentation, implementation and testing, to further enhance the functionality of the MR application, as *"our imagination is the only limit to what we can hope to have in the future."*

References

[1] M. Abraham and M. Annunziata. Augmented reality is already improving worker performance. *Harvard Business Review*, 2017. URL https://hbr.org/2017/03/augmented-reality-is-already-improving-worker-performance.

[2] AIIC. "Ilmatar" crane - OPC UA python library. 2019. URL https://github.com/AaltoIIC/ilmatar-python-lib.

[3] AIIC. "Ilmatar" crane - GraphQL API for OPC UA servers. 2019. URL https://github.com/AaltoIIC/OPC-UA-GraphQL-Wrapper.

[4] M. Billinghurst, A. Clark, and G. Lee. A survey of augmented reality. *Foundations and Trends in Human-Computer Interaction*, 8(2-3):73–272, 2014. ISSN 15513963. doi: 10.1561/1100000049.

[5] Y. Cai, B. Starly, P. Cohen, and Y. S. Lee. Sensor Data and Information Fusion to Construct Digital-twins Virtual Machine Tools for Cyber-physical Manufacturing. *Procedia Manufacturing*, 10:1031–1042, 2017. ISSN 23519789. doi: 10.1016/j.promfg.2017.07.094. URL http://dx.doi.org/10.1016/j.promfg.2017.07.094.

[6] D. Cearley and B. Burke. Identifies the top 10 strategic technology trends for 2019. pages 5–9, 2018. URL https://www.gartner.com/en/newsroom/press-releases/.

[7] R. Drath and A. Horch. Industrie 4.0: Hit or hype? [Industry Forum]. *IEEE Industrial Electronics Magazine*, 8(2):56–58, 2014. ISSN 19324529. doi: 10.1109/MIE.2014.2312079.

[8] R. T. Fielding and R. N. Taylor. *Architectural Styles and the Design of Network-Based Software Architectures*. University of California, Irvine, 2000. ISBN 0599871180. AAI9980887.

[9] K. Främling, J. Holmström, T. Ala-Risku, and M. Kärkkäinen. Product agents for handling information about physical objects. *Report of Laboratory of information processing science series B*, 153(3):1–20, 2003. URL http://www.cs.hut.fi/Publications/Reports/B153.pdf.

[10] D. Gorecky, M. Schmitt, M. Loskyll, and D. Zühlke. Human-machine-interaction in the industry 4.0 era. In *2014 12th IEEE International Conference on Industrial Informatics (INDIN)*, pages 289–294, 2014.

[11] GraphQL. GraphQL specification versions. 2020. URL https://spec.graphql.org/.

[12] M. Grieves. Digital twin: Manufacturing excellence through virtual factory replication. 03 2015. URL https://www.researchgate.net/publication/275211047_Digital_Twin_Manufacturing_Excellence_through_Virtual_Factory_Replication.

[13] S. Grüner, J. Pfrommer, and F. Palm. A RESTful extension of OPC UA. *2015 IEEE World Conference on Factory Communication Systems (WFCS)*, pages 1–4, 2015.

[14] S. Haag and R. Anderl. Digital twin – proof of concept. *Manufacturing Letters*, 15(February):64–66, 2018. ISSN 22138463. doi: 10.1016/j.mfglet.2018.02.006. URL https://doi.org/10.1016/j.mfglet.2018.02.006.

[15] J. Hietala. Real-time two-way data transfer with a Digital Twin via web interface. *Master's thesis, Aalto University*, 2020. URL https://aaltodoc.aalto.fi/handle/123456789/43524?locale-attribute=fi.

[16] P. Hublikar. A prototype of a digital twin with mixed reality and voice user interfaces for controlling a smart industrial crane. *Master's thesis, Aalto University*, 2019. URL https://aaltodoc.aalto.fi/handle/123456789/42792?locale-attribute=en.

[17] A. Juuso. Platform for industrial internet and digital twin focused education, research, and innovation: Ilmatar the overhead crane. *IEEE World Forum on Internet of Things, WF-IoT 2018 - Proceedings*, 2018-Janua, 2018. URL https://ieeexplore.ieee.org/document/8355217.

[18] P. Keitler. Management of tracking: and tracking accuracy in industrial augmented reality environments. *Dissertation, Technical University of Munich*, 2011. URL https://mediatum.ub.tum.de/node?id=1007293.

[19] KoneCranes. Feature description - AIIC crane OPC UA control interface (not published). (2007):1–6, 2019.

[20] KoneCranes. Feature description - AIIC crane OPC UA interface (not published). (2007):1–8, 2019.

[21] D. Linka. GraphQL vs REST. *Medium*, 2018. URL https://medium.com/@LinkaDaisy/graphql-vs-rest-9ad144204b22.

[22] Microsoft. .NET fundamentals - task parallel library. 2017. URL https://docs.microsoft.com/en-us/dotnet/standard/parallel-programming/task-parallel-library-tpl.

[23] Microsoft. .NET fundamentals - data parallelism (task parallel library). 2017. URL https://docs.microsoft.com/en-us/dotnet/standard/parallel-programming/data-parallelism-task-parallel-library.

[24] Microsoft. Case study - my first year on the hololens design team. 2018. URL https://docs.microsoft.com/en-us/windows/mixed-reality/out-of-scope/case-study-my-first-year-on-the-hololens-design-team.

[25] Microsoft. Parallel programming in .NET. 2018. URL https://docs.microsoft.com/en-us/dotnet/standard/parallel-programming/.

[26] Microsoft. MRTK documentation - input overview. 2019. URL https://microsoft.github.io/MixedRealityToolkit-Unity/Documentation/Input/Overview.html.

[27] Microsoft. Hololens (1st gen) hardware. 2019. URL https://docs.microsoft.com/en-us/hololens/hololens1-hardware.

[28] Microsoft. Gaze and commit. 2019. URL https://docs.microsoft.com/en-us/windows/mixed-reality/gaze-and-commit.

[29] Microsoft. MRTK - Mixed Reality Toolkit documentation, interactable. 2020. URL https://microsoft.github.io/MixedRealityToolkit-Unity/Documentation/README_Interactable.html.

[30] Microsoft. Hololens 1 gestures for authoring and navigating in dynamics 365 guides. 2020. URL https://docs.microsoft.com/en-us/dynamics365/mixed-reality/guides/authoring-gestures.

[31] P. Milgram, H. Takemura, A. Utsumi, and F. Kishino. Augmented reality: a class of displays on the reality-virtuality continuum. *Telemanipulator and Telepresence Technologies*, 2351(December 2013):282–292, 1995.

[32] E. Negri, L. Fumagalli, and M. Macchi. A Review of the Roles of Digital Twin in CPS-based Production Systems. *Procedia Manufacturing*, 11(June): 939–948, 2017. ISSN 23519789. doi: 10.1016/j.promfg.2017.07.198. URL http://dx.doi.org/10.1016/j.promfg.2017.07.198.

[33] M. R. Nichols. How augmented reality will disrupt the manufacturing industry. 2019. URL https://blog.thomasnet.com/augmented-reality-manufacturing.

[34] OPC-Foundation. What is OPC? 2015. URL https://opcfoundation.org/about/what-is-opc/.

[35] OPC-Foundation. OPC Foundation(2017a) OPC UA part 1: Overview and concepts. 2017. URL https://opcfoundation.org/developer-tools/specifications-unified-architecture.

[36] OPC-Foundation. OPC Foundation (2017c) OPC UA part 4: Services. 2017. URL https://opcfoundation.org/developer-tools/specifications-unified-architecture.

[37] A. Parrott and L. Warshaw. Industry 4.0 and the digital twin. *Deloitte University Press*, pages 1–17, 2017. URL https://dupress.deloitte.com/dup-us-en/focus/industry-4-0/digital-twin-technology-smart-factory.html.

[38] M. E. Porter and J. E. Heppelmann. How smart, connected products are transforming competition. *Harvard Business Review*, (November 2014), 2014. ISSN 00178012. URL https://hbr.org/2014/11/how-smart-connected-products-are-transforming-competition?utm_source=digg.

[39] D. Pustka. Construction of data flow networks for augmented reality applications". *Proc. Dritter Workshop Virtuelle und Erweiterte Realität der GI-Fachgruppe VR/AR, Koblenz, Germany*, 2006.

[40] D. Pustka, M. Huber, M. Bauer, and G. Klinker. Handling error in Ubiquitous tracking setups. *Master's thesis. TU München, Munich, Germany*, 2004.

[41] D. Pustka, M. Huber, M. Bauer, and G. Klinker. Spatial relationship patterns: Elements of reusable tracking and calibration systems. *Proc. IEEE International Symposium on Mixed and Augmented Reality (IS- MAR'06)*, 2006.

[42] D. Pustka, M. Huber, F. Echtler, and P. Keitler. UTQL: the Ubiquitous tracking query language v1.0. *Tech. Rep. TUM-I0718, Institut für Informatik, Technische Universität München*, 2007.

[43] T. Reicher. A framework for dynamically adaptable augmented reality systems. *Dissertation, Technische Universität München*, 2004. URL https://mediatum.ub.tum.de/601755.

[44] RestSharp. Restsharp homepage. 2020. URL https://restsharp.dev/.

[45] L. Richardson and S. Ruby. *RESTful web services*. Farnham: O'Reilly, 2017. ISBN 9780198520115.

[46] M. H. Rüßmann, Michael; Markus Lorenz, Philip Gerbert, Manuela Waldner, Jan Justus, Pascal Engel. Industry 4.0: The Future of Productivity and Growth in Manufacturing Industries. *Boston Consulting*, pages 1–14, 2015. ISSN 9783935089296. doi: 10.1007/s12599-014-0334-4.

[47] M. Shafto, M. Conroy, R. Doyle, E. Glaessgen, C. Kemp, J. LeMoigne, and L. Wang. Modeling, simulation, information technology & processing roadmap - technology area 11. *National Aeronautics and Space Administration*, page 27.

[48] M. Speicher, B. D. Hall, and M. Nebeling. What is mixed reality? In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, CHI '19, page 1–15, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450359702. doi: 10.1145/3290605.3300767. URL https://doi.org/10.1145/3290605.3300767.

[49] A. Syberfeldt, O. Danielsson, M. Holm, and T. Ekblom. Augmented reality at the industrial shop-floor. In L. T. De Paolis and A. Mongelli, editors, *Augmented and Virtual Reality*, pages 201–209, Cham, 2014. Springer International Publishing. ISBN 978-3-319-13969-2.

[50] R. Tsai. A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses. *IEEE Journal of robotics and Automation*, 1987.

[51] Unity. Unity documentation - MonoBehaviour. 2019. URL https://docs.unity3d.com/ScriptReference/MonoBehaviour.html.

[52] Unity. What is multithreading? 2019. URL https://docs.unity3d.com/Manual/JobSystemMultithreading.html.

[53] Vuforia. Vuforia developer library - image targets. 2019. URL https://library.vuforia.com/features/images/image-targets.html.

[54] Vuforia. Vuforia developer library - Vuforia fusion. 2020. URL https://library.vuforia.com/content/vuforia-library/en/articles/Training/vuforia-fusion-article.html.

[55] Vuforia. Vuforia developer library - device tracking overview. 2020. URL https://library.vuforia.com/features/environments/device-tracker-overview.html.

[56] Vuforia. Developing Vuforia Engine apps for HoloLens. 2020. URL https://library.vuforia.com/content/vuforia-library/en/articles/Training/Developing-Vuforia-Apps-for-HoloLens.html.

[57] Vuforia. Vuforia engine developer portal. 2020. URL https://developer.vuforia.com/.

[58] N. Yadav. Understanding display techniques in augmented reality. 2018. URL https://blog.prototypr.io/understanding-display-techniques-in-augmented-reality-c258b911b5c9.

[59] Z. Zhu, C. Liu, and X. Xu. Visualisation of the digital twin data in manufacturing by using augmented reality. *Procedia CIRP*, 81:898–903, 2019. ISSN 22128271. doi: 10.1016/j.procir.2019.03.223. URL https://doi.org/10.1016/j.procir.2019.03.223.